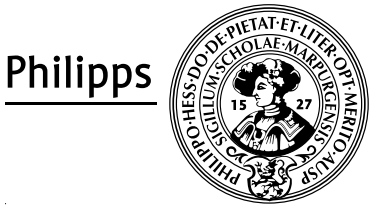

Learning Nonlinear Monotone Classifiers Using The Choquet Integral

Dissertation

zur Erlangung des Doktorgrades
der Naturwissenschaften
(Dr. rer. nat.)

dem Fachbereich Mathematik und Informatik
der Philipps-Universität Marburg
vorgelegt

von
Ali Fallah Tehrani

Marburg, 2014

Vom Fachbereich Mathematik und Informatik
der Philipps-Universität Marburg als Dissertation
angenommen am: 17.06.2014

Erstgutachter: Prof. Dr. rer. nat. Eyke Hüllermeier
Zweitgutachter: Dr. Krzysztof Dembczyński
Tag der mündlichen Prüfung: 20.06.2014

To my parents

تقدیم به بهترین ها

Contents

List of Figures

List of Tables

Acknowledgment

This thesis would not have been completed, without support and help from a many people. I would like to take this opportunity to appreciate the academics as well my family and my friends.

In this regard, first and foremost I would like to express my sincere appreciation to my supervisor, Prof. Dr. Eyke Hüllermeier, who acquainted me with a new field interdisciplinary computer science, statistics and mathematics. During my PhD, Prof. Hüllermeier has oriented me by proposing numerous great ideas and perspectives. His logical way of thinking always improved notably my initial ideas, and specifically his advices enhanced the structure of this dissertation. I am as well deeply grateful. For me, coming from the theoretical informatics and mathematics community, this was an ideal opportunity to discover and to use sound mathematical and statistical tools to solve real challenging problems from an artificial intelligence point of view. I am very thankful to him for giving me the opportunity to explore this area and support me during the PhD.

Also I would like to take this opportunity to thank my collaborators as well my colleagues at Philipps Universität Marburg. From them I learned many useful ideas. In this respect, I would like to thank Dr. Krzysztof Dembczyński, Dr. Christophe Labreuche, Dr. Marc Strickert, Dr. Thomas Fober, Dr. Weiwei Cheng, Dr. Robert Busa-Fekete, Maryam Nassiri, Ammar Shaker, Robin Senge, Amira Abdel-Aziz, Florian Meyer, Michael Bräuning, Dr. Willem Waegeman, Sascha Henzgen, Dr. Marco Mernberger, Dirk Schäfer, Manish Agarwal, Dr. Anne Knöller, Patrice Schlegel and Dr. Hyung won Koh.
Especially I appreciate the efforts of Krzysztof, Marc, Thomas and Weiwei.

I owe so much to my family and my friends for supporting me during my education, specifically my PhD. My special thanks and appreciations go to my parents and my sister, who during this time interval were considerate and always giving

positive energy and encouraging me to my research and my PhD. Nina, I am deeply indebted to you for everything that you done for the family.

Summary

The learning of predictive models that guarantee a monotonic relationship between the output (response) and input (predictor) variables has received increasing attention in machine learning in recent years. While being less problematic for linear models, the difficulty of ensuring monotonicity increases with the flexibility of the underlying model class.

This thesis advocates the so-called Choquet integral as a mathematical tool for learning monotone nonlinear models for classification. While being widely used as a flexible aggregation function in fields such as multiple criteria decision making, the Choquet integral is much less known in machine learning so far. Apart from combining monotonicity and flexibility in a mathematically sound and elegant manner, the Choquet integral has additional features making it attractive from a machine learning point of view. For example, it offers measures for quantifying the importance of individual predictor variables and the interaction between groups of variables, thereby supporting the interpretability of a model.

Concrete methods for learning with the Choquet integral are developed on the basis of two different approaches, namely maximum likelihood estimation and structural risk minimization. While the first approach leads to a generalization of logistic regression, the second one is put into practice by means of support vector machines. In both cases, the learning problem essentially comes down to identifying the fuzzy measure on which the Choquet integral is defined. Since this measure has a large number of degrees of freedom, learning the Choquet integral is critical not only from a complexity point of view but also with regard to proper generalization. Therefore, both methods are analyzed theoretically, and different approaches to regularization and complexity reduction are proposed.

Experimental results conducted on a set of suitable benchmark data are quite promising and suggest that the combination of monotonicity and flexibility offered by the Choquet integral facilitates strong performance in practical applications.

Zusammenfassung

In der jüngeren Vergangenheit hat das Lernen von Vorhersagemodellen, die eine monotone Beziehung zwischen Ein- und Ausgabevariablen garantieren, wachsende Aufmerksamkeit im Bereich des maschinellen Lernens erlangt. Besonders für flexible nichtlineare Modelle stellt die Gewährleistung der Monotonie eine große Herausforderung für die Umsetzung dar.

Die vorgelegte Arbeit nutzt das Choquet Integral als mathematische Grundlage für die Entwicklung neuer Modelle für nichtlineare Klassifikationsaufgaben. Neben den bekannten Einsatzgebieten des Choquet-Integrals als flexible Aggregationsfunktion in multi-kriteriellen Entscheidungsverfahren, findet der Formalismus damit Eingang als wichtiges Werkzeug für Modelle des maschinellen Lernens. Neben dem Vorteil, Monotonie und Flexibilität auf elegante Weise mathematisch vereinbar zu machen, bietet das Choquet-Integral Möglichkeiten zur Quantifizierung von Wechselwirkungen zwischen Gruppen von Attributen der Eingabedaten, wodurch interpretierbare Modelle gewonnen werden können.

In der Arbeit werden konkrete Methoden für das Lernen mit dem Choquet Integral entwickelt, welche zwei unterschiedliche Ansätze nutzen, die Maximum-Likelihood-Schätzung und die strukturelle Risikominimierung. Während der erste Ansatz zu einer Verallgemeinerung der logistischen Regression führt, wird der zweite mit Hilfe von Support-Vektor-Maschinen realisiert. In beiden Fällen wird das Lernproblem im Wesentlichen auf die Parameter-Identifikation von Fuzzy-Maßen für das Choquet Integral zurückgeführt. Die exponentielle Anzahl von Freiheitsgraden zur Modellierung aller Attribut-Teilmengen stellt dabei besondere Herausforderungen im Hinblick auf Laufzeitkomplexität und Generalisierungsleistung. Vor deren Hintergrund werden die beiden Ansätze praktisch bewertet und auch theoretisch analysiert. Zudem werden auch geeignete Verfahren zur Komplexitätsreduktion und Modellregularisierung vorgeschlagen und untersucht.

Die experimentellen Ergebnisse sind auch für anspruchsvolle Referenzprobleme im Vergleich mit aktuellen Verfahren sehr gut und heben die Nützlichkeit der Kombination aus Monotonie und Flexibilität des Choquet Integrals in verschiedenen An-

sätzen des maschinellen Lernens hervor.

List of Abbreviations

AI	artificial intelligence
CI	Choquet integral
ELECTRE	elimination and choice expressing reality
I.I.D.	identically independently distributed
LR	logistic regression
LL	log-linear models
MCDM	multiple criteria decision making
MLE	maximum likelihood estimation
MUTA	multi attribute utilities theory
OCR	ordinal choquistic regression
OLR	ordinal logistic regression
OWA	ordered weighted averaging
PROMETHEE	preference ranking organization method for enrichment of evaluations
QP	quadratic programming
SRM	structural risk minimization
SVM	support vector machine
SQP	sequential quadratic programming
TOPSIS	technique for order preference by similarity to ideal solution

WM	weighted mean

List of Notations

$$[K] := \{1, \ldots, K-1\}, \qquad K \in \mathbb{N}$$

$$sign(x) := \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

$$\text{(indicator function)} \quad \mathbb{I}_{\mathcal{Y}} : \mathcal{X} \to \{0, 1\}, \quad \text{where} \quad \mathcal{Y} \subset \mathcal{X}$$

$$\mathbb{I}_{\mathcal{Y}}(x) := \begin{cases} 1 & \text{if } x \in \mathcal{Y} \\ 0 & \text{otherwise} \end{cases}$$

$$\bigtimes_{i=1}^{n} S_i = S_1 \times \ldots \times S_n$$

$$F : \mathbb{R}^m \to \mathbb{R}$$

$$\nabla F = \Big(\frac{\partial F}{\partial x_1}, \ldots, \frac{\partial F}{\partial x_m} \Big)$$

$\mathcal{P}(C)$ powerset of set C

m	number of attributes
n	number of observations

There is a real danger that computers will develop intelligence and take over. We urgently need to develop direct connections to the brain so that computers can add to human intelligence rather than be in opposition. (Stephen Hawking)

1 Introduction

Machine learning as a sub field of AI attempts to generalize data by recognizing proper structures, patterns and relationships. Data plays the role of an experience or set of experiences and the ultimate goal is to design machines which are able to learn from these experiences. The machines are adapted by experiences (given data) and later on can improve themselves by adapting more experiences. In general, the task in machine learning can be characterized by unsupervised and supervised learning.

In supervised learning the task is to make a generalization based on some observations and their responses; the response of an observation can be seen as output of an unknown function given the observation. Contrary to supervised learning, in unsupervised learning the goal is to generalize the observations without any response. In fact, what distinguishes unsupervised learning from supervised learning, is the type of data. In unsupervised learning the observations do not imply any information about response, whereas in supervised learning the responses are additionally given. More concretely, the core idea in supervised learning is to generalize the dependency between observations and their responses in terms of a structure, a pattern, a relationship or a function.

As will be clear later on, types of data (experiences), prior knowledge and the learning algorithm have strong influences on such generalizations. Therefore the

interaction of selecting a learning algorithm concerning the type of data and prior knowledge is a crucial point and can improve the precision of generalization. In the next section, the basic idea of prior knowledge is presented.

1.1 Prior Knowledge

In order to generalize data in a more accurate way, the machine can use some promising knowledge. This knowledge indicates some trustworthy properties or relationships with respect to observations. According to the existence of prior knowledge, the space of candidate solutions is restricted to a sub space, in which such dependencies or relationships are always valid. Considering the existence of prior knowledge, there is a chance to improve inference.

For instance assume there is a generator which generates randomly numbers between $[0, 1]$ using a Gaussian distribution with mean (μ) equal to 0.5 and variance (σ) equal to 0.01. In this case, taking this distribution into account, the numbers close to 0 or 1 are barely expected. Such information can be seen as prior knowledge.

1.2 Monotonicity as a Specific Type of Prior Knowledge

In this section, a specific kind of dependency between observations and their responses is discussed. This dependency in many applications is indeed desirable, and therefore has attracted considerable attention in general and in particular in machine learning applications.
Before going into details, the concept of *pareto dominance* should be introduced:

Suppose $\boldsymbol{x} = (x_1, \ldots, x_m)$ and $\boldsymbol{x}^* = (x_1^*, \ldots, x_m^*)$ are two elements in $\mathbb{R}^m$. The element $\boldsymbol{x}^*$ is said dominates element $\boldsymbol{x}$, in terms of pareto ($\boldsymbol{x} \preceq \boldsymbol{x}^*$), if

$$x_i \leq x_i^*, \qquad \forall i \quad 1 \leq i \leq m$$

In order to emphasize this $\preceq$ is a Pareto dominance relation, in this thesis, $\preceq_P$ is used.

Now assume function $f : \mathcal{X}_1 \times \ldots \times \mathcal{X}_m \to \mathbb{R}$, where $\mathcal{X}_1 \times \ldots \times \mathcal{X}_m \subset \mathbb{R}^m$, is given. The function $f(\cdot, \ldots, \cdot)$ is said to be a monotone function, if

$$\forall \boldsymbol{x}, \boldsymbol{x}^* \in \mathcal{X}_1 \times \ldots \times \mathcal{X}_m \text{ s.t. } \boldsymbol{x} \preceq \boldsymbol{x}^* \quad \text{then} \quad f(\boldsymbol{x}) \leq f(\boldsymbol{x}^*)$$

This relationship between the domain of $f(\cdot, \ldots, \cdot)$ and range of $f(\cdot, \ldots, \cdot)$ is called monotonicity dependency. In a general case, suppose $\mathcal{D} = \left\{(\boldsymbol{x}_i, y_i)\right\}_{i=1}^n \subset \mathbb{R}^m \times \mathbb{R}$ be given data, where $\left\{\boldsymbol{x}_i\right\}_{i=1}^n$ are n observations and $\left\{y_i\right\}_{i=1}^n$ are their responses. The data $\mathcal{D}$ is said to be monotone, if

$$\forall \boldsymbol{x}_i,\ \boldsymbol{x}_j \in \mathcal{D} \text{ s.t. } \boldsymbol{x}_i \preceq \boldsymbol{x}_j \quad \text{then} \quad y_i \leq y_j$$

In general, the response set can be considered as an ordinal set. This issue will be discussed in more details in Chapter 4.

Since the monotonicity dependency demonstrates a kind of relationship. From a supervised learning point of view, monotonicity is therefore counted as prior knowledge.

The following are some examples related to real applications:

1.2.1 Medicine

Suppose an expert wants to model the dependency between a heart attack and human factors. It is obvious that the heart attack depends on several factors. For instance, a heart attack depends on high blood pressure, tobacco consumption, age, weight, etc. In this case, there is obviously a direct dependency between these factors and the probability of a heart attack occurring. For instance in the case of the age factor; the higher one's age is, the higher the probability of the heart attack occurring. Such information can be considered as a kind of background knowledge.

1.2.2 Buying a Car

From a user's point of view, a car can be characterized by several factors. For instance, the user is usually interested in the engine power, capacity of the car, the size of car boot, the safety level of car and the maintenance costs. In addition, suppose the price of the car is given. Obviously there is a direct relationship between the mentioned factors and the price of the car. The better the factors are, the higher the price. This dependency indeed is the monotonicity dependency.

1.3 Monotonicity and Multiple Criteria Decision Making

As mentioned earlier, monotonicity is an enticing property and in many applications is required. From this perspective, one natural question is: what kind of dependency should be taken into consideration to satisfy this expectation, namely, which framework can assure monotonicity dependency. To this end, *multiple criteria decision making* (MCDM) provides a family of monotone functions, i.e., each of which can assure monotonicity property. Hence from a monotonicity point of view, the generalization underlying such functions fulfills our expectations. From a *multiple criteria decision making* point of view, those functions have a specific name; aggregation function. A comprehensive review about MCDM is given in Chapter 3.

1.3.1 The Choquet Integral

As mentioned in previous section, the MCDM serves a family of monotone functions. Seen from this perspective, each can satisfy monotonicity properties. In this regard, it is worth mentioning that a precise generalization is always desirable, however what makes the generalization more understandable is the ability to interpret the generalization in a promising way.
Among the monotone functions in this family, the so called *Choquet integral* satisfies these expectations; it is a monotone function and it is interpretable as well. In addition, the *Choquet integral* is a non-linear function. The non-linearity yields the ability to capture non-linear dependency. We discuss this issue in greater details in Chapters 3 and 4.

1.4 The Choquet Integral and its Contribution to Machine Learning

The mentioned properties make the Choquet integral more desirable to exploit it in the machine learning field; specifically for monotone learning. So far the Choquet integral has been taken into account as a powerful aggregation function in decision theory and multiple criteria decision making [42, 44, 54]. It has been used in many applications, e.g., selecting an alternative or ordering several alternatives. However, it has not been widely used in the machine learning field for arbitrary data. In general, the proper parameters for the Choquet integral are given by experts in the related fields. Seen from this view, there are at least two disadvantages:

- For each data field, an expert is required, who can at least guess the proper parameters. Especially if the data is unknown, it is impossible even to have an approximate solution.
- For the large number of observations, it is almost impossible to find the parameters even approximately.

Already mentioned, the Choquet integral yields several promising properties, which can make the generalization more understandable. Due to its non-linearity, there is also a chance to model more complicated dependencies. As can be seen, there is certainly a need to utilize the Choquet integral for arbitrary data. To this end, the core idea of this thesis is to embed the Choquet integral in a machine learning framework. From a machine learning point of view, this thesis embeds the Choquet integral into two different frameworks, namely, the probabilistic and deterministic frameworks. For the probabilistic framework, the *Maximum Likelihood Approach*, and for deterministic framework *Kernel Based Learning* and *Support Vector Machines*, are taken into account.
For each framework, the precise algorithms and the core motivations are given. Also the advantages and disadvantages of each of them are described in a comprehensive manner.

2

Background in Machine Learning

2.1 Introduction

As already mentioned, monotonicity describes a kind of dependency between observations and their responses. Thus, it is related to a supervised learning problem. From this perspective, this chapter begins by exploring some preliminaries in supervised learning. In this regard, the basic ideas and definitions of the learning problem, loss functions and example of classic learning problems are presented. In Section 2.3, two different approaches for induction, namely the *maximum likelihood principle* and *structural risk minimization* are introduced. In Section 2.4, the linear logistic regression and support vector machines, which are derived from inductive principles, are introduced. Finally, the idea of monotone classifiers is presented.

2.2 Supervised Learning

In supervised machine learning, the final goal is to induce a model from the observations and their responses. The observations used for induction are called training data. They are defined based on some attributes, e.g., weight, height or consumption. More formally an observation/instance $\boldsymbol{x}$ has the following form:

$$\boldsymbol{x} = (x_1, \ldots, x_m) \in \mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_m \ ,$$

where $\mathcal{X}_i$ is domain of attributes i-th.

Because of consistency, a well-known assumption is taken to the account called *i.i.d* (independent and identically distributed). This assumption assures that the data points are drawn independent and identically.

Before continuing the topic the following definition should be introduced: Assume the joint probability $\mathbf{P}(\cdot, \cdot)$ on $\mathcal{X} \times \mathcal{Y}$ is given, where $\mathcal{X}$ is domain of attributes and $\mathcal{Y}$ is domain of responses. Then given an observation $\boldsymbol{x}$, the response y, for joint probability distribution $\mathbf{P}(\boldsymbol{x}, y)$ is called *ground truth*.

2.2.1 Basic Setting

In order to make a proper generalization from given training data, all candidates for generalization are taken into account. The set of all candidates for generalization is called the *hypothesis space*. The *hypothesis space* is defined formally as follows:

$$\mathcal{H} = \Big\{ h \Big| h : \mathcal{X} \to \mathcal{Y} \Big\}$$

Here $\mathcal{X}$ is domain of attributes and $\mathcal{Y}$ is domain of responses. Also every function $h(\cdot)$ in the hypothesis space is called a hypothesis. Additionally, if the set $\mathcal{H}$ is restricted to a specific family of hypothesis $\mathcal{F}$, it is called the model class under $\mathcal{F}$ and is defined as follows:

$$\mathcal{H}_{\mathcal{F}} = \Big\{ h \in \mathcal{F} \Big| h : \mathcal{X} \to \mathcal{Y} \Big\}$$

From a probabilistic point of view, the risk of the hypothesis h, namely, $R(h)$ is defined in terms of its expected loss:

$$R(h) = \int_{\mathcal{X} \times \mathcal{Y}} l(h(\boldsymbol{x}), y) d\mathbf{P}_{XY}(\boldsymbol{x}, y) \ , \tag{2.1}$$

where $l(\cdot, \cdot)$ is a loss function penalizing incorrect predictions. The final goal is to find a hypothesis which minimizes the risk function. In the following sections related to each learning problem, the corresponding loss function is introduced.

2.2.2 Loss Functions

From a machine learning point of view, given the training data, the goal is to induce a model which is in agreement with ground truth as much as possible. However, quantifying such agreement is defined in a completely different way. In fact, from a machine learning point of view, we are interested indeed in disagreement, namely, how many mistakes the prediction has. Such determinations are called loss functions and depends on the problem, the algorithm tries to minimize through the expectation of loss functions (risk function). Taking this fact into account, respect to each problem, the risk function can demonstrate the performance of the model and in essence is defined related to learning-problem. The algorithm in accordance with the problem minimizes such risk function.

Since the number of training data is restricted, the whole learning-space cannot be covered. What is expected is to approximate the loss function. In this regard, two types of loss functions can be considered from the scholarly literature as follows:

- **Empirical Risk Minimization (ERM):** In the case of supervised learning, the empirical loss function refers to the case, when the joint probability distribution of the inputs and outputs is unknown. However, there are some observations (training examples) through which the error approximatively computed.
- **Theoretical Risk Minimization (TRM):** Knowing the joint probability distribution with respect to inputs and outputs, the exact risk (error) can be computed. This error is called theoretical loss function.

Note that usually it is not possible to find the exact joint probability distribution. Hence, it is common the empirical risk minimization to be taken into account.

$0/1$ Loss

In the binary class classification problem, the most commonly used loss is the simple $0/1$ loss given by:

$$l_{0/1}(y, \hat{y}) = \begin{cases} 0 & \text{if } \hat{y} = y \\ 1 & \text{if } \hat{y} \neq y \end{cases}, \tag{2.2}$$

in which y is ground truth and $\hat{y}$ is predicted label. In order to have normalized version, the following form is usually used:

$$l^*_{0/1} = \frac{1}{n}\sum_{i=1}^{n} l(y_i, \hat{y}_i) \ .$$

Therefore $l^*_{0/1}$ ranges in $[0, 1]$.

L_1 Loss

The L_1 loss or say Manhattan distance is defined as follows:

$$L_1(y, \hat{y}) = |y - \hat{y}| \tag{2.3}$$

in which y is ground truth and $\hat{y}$ is predicted label. Moreover for $y, \hat{y}$ the assumption is $y, \hat{y} \in \{1, \ldots, K\}$. L_1 loss can be used for the ordinal classification problem. The normalized version of the L_1 loss usually is assumed in the following:

$$L^*_1 = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \ .$$

Hence L^*_1 ranges in $[0, K-1]$.

2.2.3 Binary Classification

From a supervised learning point of view, binary classification is a kind of rudimentary learning problem. In this case, every instance is labeled by a label from $\{-1, +1\}$. So now assume the training data is given as follows:

$$\mathcal{D} = \Big\{(\boldsymbol{x}_i, y_i)\Big\}_{i=1}^{n} \subset \mathcal{X} \times \Big\{-1, +1\Big\} \ ,$$

in which $\mathcal{D}$ is supposed to be an *i.i.d.* (independent and identically distributed) generated by an underlying (though unknown) probability measure $\mathbf{P}_{XY}$ on $\mathcal{X} \times \mathcal{Y}$. The goal in binary classification is to induce a classifier $\mathcal{L} : \mathcal{X} \to \{-1, +1\}$, which minimizes the corresponding risk function. In this case, the $0/1$ loss, is the most commonly used loss.

2.2.4 Ordinal Classification

In binary classification, the response consists of only two classes, typically called the negative (-1) and the positive $(+1)$ class. In ordinal classification, the response contains more classes, where in addition the classes are ordered. More formally, assume $\mathcal{Y} = \{y_1, \ldots, y_K\}$ are the classes. In an ordinal case, it is supposed that,

$$y_{\sigma(1)} \prec y_{\sigma(2)} \prec \ldots \prec y_{\sigma(K)} \ ,$$

where σ is a permutation of $\{1, \ldots, K\}$ and $\prec$ is referred to as an ordering.
In this thesis, the ordinal classes are natural numbers and have the following form:

$$y_1 < y_2 < \ldots < y_K \ ,$$

The goal in ordinal classification is to learn a classifier $\mathcal{L} : \mathcal{X} \to \mathcal{Y}$ from a given set of training data:

$$\mathcal{D} = \Big\{ (\boldsymbol{x}_i, y_i) \Big\}_{i=1}^{n} \subset \mathcal{X} \times \mathcal{Y} \ .$$

The data $\mathcal{D}$ is supposed to be an *i.i.d.* sample generated by an underlying (though unknown) probability measure $\mathbf{P}_{XY}$ on $\mathcal{X} \times \mathcal{Y}$. A common goal, then, is to induce a classifier with minimal risk, where the risk $R(\mathcal{L})$ of a classifier $\mathcal{L}$ is defined in terms of its expected loss, i.e., the loss in (2.1). In order to take the order of the classes in an ordinal classification case, usually L_1 loss is taken into account.

2.3 The Principles of Induction

The principles of induction are used to find an optimal generalization from seen observations. In this case, assume data and also the hypothesis space are given. In fact, the duty of inductive principles is to chose a proper hypothesis in the hypothesis space.

2.3.1 Maximum Likelihood Estimation (MLE)

Maximum likelihood estimation makes induction in a probabilistic frame work. Roughly speaking, maximum likelihood maximizes the likelihood of observing data. Accordingly, it seeks out the parameters, which are most likely for the given

data. Historically at the beginning of 20th century Fisher proposed the initial idea of maximum likelihood [1].

Following the core idea of maximum likelihood, assume the observations $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^n$ in which the *i.i.d.* assumption is supposed, are given. Moreover assume a family of density function $\{f(\boldsymbol{x};\theta)\}_\theta$ is given, where additionally the probability distribution is assumed with respect to θ is continuous. Since the *i.i.d.* assumption is supposed, the joint density function for $\{\boldsymbol{x}_i\}_{i=1}^n$ is equal to:

$$f(\boldsymbol{X};\theta) = \prod_{i=1}^{n} f(\boldsymbol{x}_i;\theta) \ .$$

From a statistical point of view, the function $\mathcal{L}(\theta;\boldsymbol{x}) = f(\boldsymbol{x};\theta)$, is called a likelihood function. Taking this fact into account, the following equality is obtained:

$$\mathcal{L}(\theta;\boldsymbol{X}) = \prod_{i=1}^{n} f(\boldsymbol{x}_i;\theta) \ .$$

In general the basic idea of maximum likelihood is to find parameter θ which maximizes the above inequality, i.e., maximizing the likelihood of observing data. Concretely the goal is to find the θ^* as follows:

$$\theta^* = \arg\max_{\theta} \mathcal{L}(\theta;\boldsymbol{X}) \ .$$

From a computational point of view, it is more convenient to maximize the logarithm of the likelihood function. To this end, the following equation is taken into account:

$$\log \mathcal{L}(\theta;\boldsymbol{X}) = \sum_{i=1}^{n} \log f(\boldsymbol{x}_i;\theta) \ .$$

Then the ultimate goal is to find parameter θ which maximizes the logarithm of the likelihood function. Interestingly, Wald in 1949 showed the consistency of maximum likelihood principle [110], namely assume ω is a closed subset of the parameter space $\Omega \setminus \{\theta^*\}$. Moreover assume $\theta_n^* = \arg\max_\theta \mathcal{L}\big(\theta;\{\boldsymbol{x}\}_{i=1}^n\big)$, and θ^* is equal to $\lim_{n\to\infty}\theta_n^*$. Then

$$\mathbf{P}\left\{\lim_{n\to\infty} \frac{\sup_{\theta\in\omega}\prod_{i=1}^n f(\boldsymbol{x}_i,\theta)}{\prod_{i=1}^n f(\boldsymbol{x}_i,\theta^*)} = 0\right\} = 1 \ .$$

It means with probability 1 the sequence $\{\theta_i\}_i$ converges to the optimal solution.

2.3.2 Structural Risk Minimization (SRM)

VC Dimension

It is clear that each hypothesis has specific properties, and in general each hypothesis space provides different properties. In this regard, the flexibility (capacity) of each hypothesis space can be studied. Here the flexibility of a hypothesis space can be seen as the ability to provide flexible hypotheses. In order to quantify the so-called capacity of one classifier Vapnik proposed in [108] the concept of the VC dimension. Roughly speaking, the VC dimension is the maximum number of instances, in which the classifier can classify those instances with respect to arbitrary labels without any mistakes. Before going into details, the concept of shattering should be introduced:

From a binary classification point of view, given n instances $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^m$ there are 2^n ways to assign labels $\{-1, +1\}$ to these instances. The instances $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ are said to be shattered by the model class $\mathcal{H}$ if, for all possible labeling (2^n cases), there exists at least one model from model class $\mathcal{H}$ which can classify the instances without any error. So the largest number of instances, which can be shattered by model class $\mathcal{H}$ is called the VC dimension of model class. More formally, based on the Vapnik and Chervonenkis, the VC dimension of model class $\mathcal{F}$ is defined as follows:

$$\max \left\{ |X| \middle| X \subset \mathcal{X},\ \forall g \in \{-1, +1\}^X, \exists h \in \mathcal{F} \text{ such that } \forall \boldsymbol{x} \in X, h(\boldsymbol{x}) = g(\boldsymbol{x}) \right\} .$$

By way of example, for a model class of linear functions with m variables, the VC dimension is equal to $m + 1$. In Figure 2.1 all existing label assignments and corresponding separations are shown. If for a model class the VC dimension is unbounded, then the VC dimension is infinite.

Loosely speaking, the VC dimension reveals the flexibility of a model class, the higher the VC dimension, the more flexible the model class is.

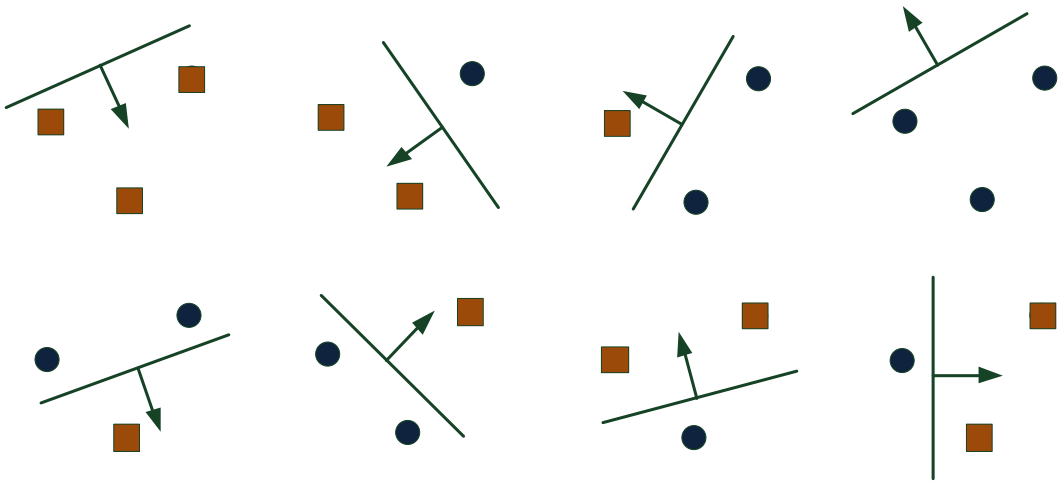

Figure 2.1: The illustration of shattering of three instances for model class linear functions with two variables

Structural Risk Minimization (SRM)

Under empirical risk minimization, two well-known problems can occur during the learning process. They are called, overfitting and underfitting. The overfitting problem refers to when the capacity (complexity) of the learner clearly is higher than what is required. Likewise, the underfitting problem occurs when the capacity (complexity) of the learner is clearly lower compared to what indeed is needed. In order to overcome this problem, or let say, find the proper learner, Vapnik proposed the idea of *structural risk minimization*. Assume a family class of learners is given. Moreover, assume there is a possibility to order the learners based on their complexity, e.g. VC dimension. The main goal under structural risk minimization is to find a trade-off between the complexity of the learner and the goodness of generalization. More formally, the goal is to minimize

$$R_{emp}(\boldsymbol{w}) + \lambda C_P(\boldsymbol{w}) \ \ ,$$

where R_{emp} is referred to the empirical risk, C_P is referred to complexity penalty and finally λ is the trade-off parameter, which is determined empirically. In this regard, Vapnik proposed a bound to show a sound dependency between the risk and empirical risk given the VC dimension of the model.

Theorem 2.1 *(Vapnik) Assume $\mathcal{H}$ is the class of functions, with a VC dimension of v. Then for any distribution P and for any sample data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ drawn from*

this distribution, the following inequality is valid with probability $1 - \eta$*.*

$$\forall h \in \mathcal{H}, \quad R(h) \leq R_{emp}(h) + \sqrt{\frac{v(\log \frac{2n}{v} + 1) - \log(\frac{n}{4})}{n}} + \frac{1}{n} \ . \tag{2.4}$$

More formally, assume there is a possibility to order the hypotheses in hypothesis space as follows:

$$\mathcal{H}_0 \subset \mathcal{H}_1 \subset \mathcal{H}_2 \subset \ldots \subset \mathcal{H} \ ,$$

where $\mathcal{H} = \cup_{i=0}^{\infty} \mathcal{H}_i$. Moreover assume the VC dimension of each $\mathcal{H}_i$ is equal to v_i. It is clear that

$$v_0 < v_1 < v_2 < \ldots$$

In general, choosing the hypothesis with a high VC dimension, due to high flexibility reduces the empirical risk, while strengthening the overfitting problem. On the other hand, choosing the hypothesis with a low VC dimension reduces the flexibility and hence increases the empirical risk. The core idea under ***SRM***, is to find a trade-off (see Figure 2.2) between the complexity of the hypothesis and the quality of fitting in order to reduce the generalization error ($R(h)$) as much as possible. Note that based on the Vapnik theorem, choosing a hypothesis with high flexibility increases the second term in the right hand side of the equation (2.4).

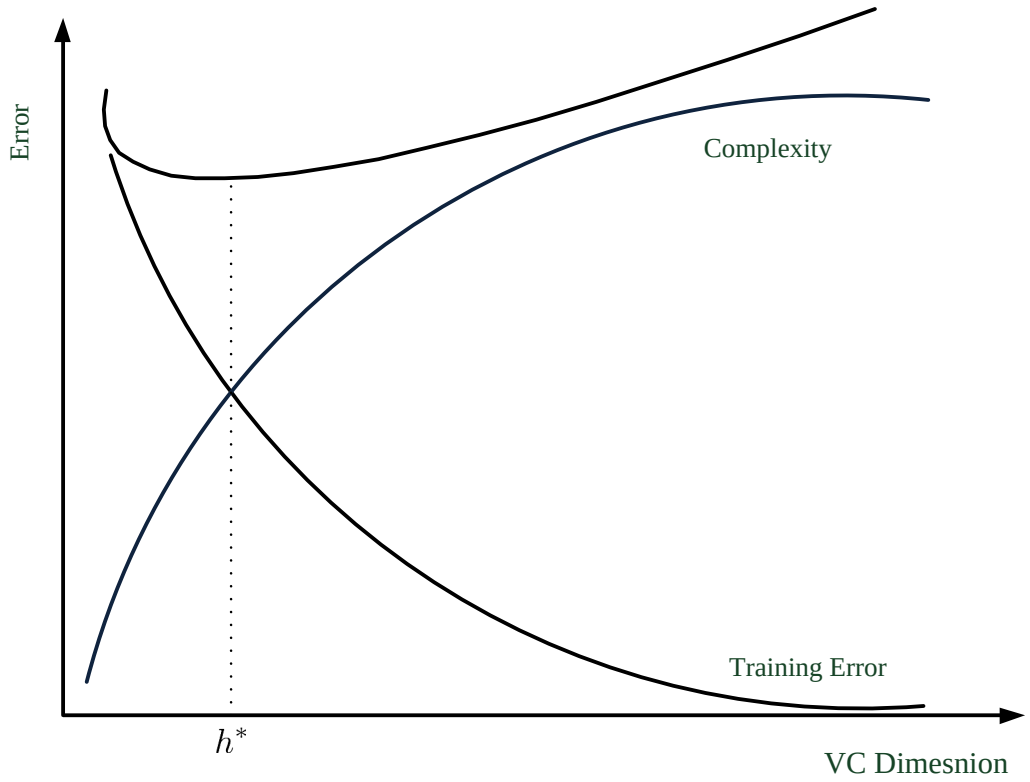

Figure 2.2: The illustration of structural risk minimization, showing the trade-off between the complexity and the quality of fitting

The Concept of Regularization

In last part it was discussed that the higher flexibility of the learner increases the chance of the overfitting problem occurring. To this end, the core idea of SRM is to find a trade-off between the complexity of the learner and the quality of generalization in a proper way. To reduce the complexity of the learner, the parameters of the learner are restricted. To this end, the idea of regularization comes into play. The idea is to consider follwoing risk:

$$R_{reg}(f) = R_{emp}(f) + \lambda\Omega(f) \ \ ,$$

where f refers to a learner. In addition, the function $\Omega(\cdot)$ measures the regularity. Here R_{reg} is called regularized risk.

2.4 The Methods Derived from Inductive Principles

2.4.1 Linear Logistic Regression

In linear regression, given the data $\mathcal{D} = \left\{(\boldsymbol{x}_i, y_i)\right\}_{i=1}^{n} \subset \mathbb{R}^m \times \mathbb{R}$, the goal is to find $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m) \in \mathbb{R}^m$ and $\epsilon \in \mathbb{R}$, such that

$$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_m x_{im} + \epsilon \qquad \forall i,\ i \in \{1, \ldots, n\}$$

Here $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{im}) \in \mathbb{R}^m$. Each x_{ij} is called a *regressor* or *predictor variable* and y_i is called a response.

The logistic regression modifies linear regression for the purpose of predicting (probabilities of) discrete classes instead of real-valued responses. To this end, the probability of the positive class (and hence of the negative class) is modeled as a linear function of the input attributes. More specifically, since a linear function does not necessarily produce values in the unit interval, the response is defined as a generalized linear model, namely in terms of the logarithm of the probability ratio:

$$\log\left(\frac{\mathbf{P}(y = 1 \,|\, \boldsymbol{x})}{\mathbf{P}(y = 0 \,|\, \boldsymbol{x})}\right) = \omega_0 + \boldsymbol{\omega}^\top \boldsymbol{x} \ , \tag{2.5}$$

where $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_m) \in \mathbb{R}^m$ is a vector of regression coefficients and $\omega_0 \in \mathbb{R}$ a constant bias (the intercept). A positive regression coefficient $w_i > 0$ means that an increase of the predictor variable x_i will increase the probability of a positive response while a negative coefficient implies a decrease of this probability. Besides, the larger the absolute value $|\omega_i|$ of the regression coefficient, the stronger the influence of x_i.
Since $\mathbf{P}(y = 0 \,|\, \boldsymbol{x}) = 1 - \mathbf{P}(y = 1 \,|\, \boldsymbol{x})$, a simple calculation yields the posterior probability

$$\pi_l \stackrel{\text{df}}{=} \mathbf{P}(y = 1 \,|\, \boldsymbol{x}) = \frac{1}{1 + \exp(-\omega_0 - \boldsymbol{\omega}^T \boldsymbol{x})} \ . \tag{2.6}$$

Assume some observations are given, where the *i.i.d.* assumption is assumed. In order to find the proper generalization for given data, suitable parameters should be determined. This can be done by employing a maximum likelihood estimation. In the following analysis we will describe in more detail the maximum likelihood estimation for a binary case. Assume the instance $\boldsymbol{x}$ and its label $y \in \{0, 1\}$ is given. Moreover assume a family of probability distribution $\mathbf{P}(\cdot, \cdot)$ is given. In general, the likelihood function for a binary class given an instance $\boldsymbol{x}$ and model parameters

$\boldsymbol{\eta}$ is equal to:

$$\mathbf{P}(y \mid \boldsymbol{x}; \boldsymbol{\eta}) = \Big\{\mathbf{P}(y = 1 \mid \boldsymbol{x}; \boldsymbol{\eta})\Big\}^{y} \cdot \Big\{\mathbf{P}(y = 0 \mid \boldsymbol{x}; \boldsymbol{\eta})\Big\}^{1-y} \tag{2.7}$$

So now, assume some observations with corresponding responses are given, where the observations are *i.i.d.* as follows:

$$\mathcal{D} = \Big\{(\boldsymbol{x}_i, y_i)\Big\}_{i=1}^{n} \subset \mathbb{R}^m \times \Big\{0, 1\Big\}$$

For more than one instance, i.e., $\big\{\boldsymbol{x}_i, y_i\big\}_{i=1}^{n}$, since *i.i.d.* is assumed, the likelihood function is formalized as follows:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\eta}) &= \mathbf{P}(\boldsymbol{y} \mid \boldsymbol{X}; \boldsymbol{\eta}) = \mathbf{P}\big(y_1, \ldots, y_n \mid \boldsymbol{x}_1 \ldots, \boldsymbol{x}_n; \boldsymbol{\eta}\big) \\
&= \mathbf{P}\big(y_1 | \boldsymbol{x}_1; \eta\big) \times \ldots \times \mathbf{P}\big(y_n | \boldsymbol{x}_n; \eta\big) \\
&= \prod_{i=1}^{n} \Big\{\mathbf{P}(y_i = 1 \mid \boldsymbol{x}_i; \boldsymbol{\eta})\Big\}^{y_i} \cdot \Big\{\mathbf{P}(y_i = 0 \mid \boldsymbol{x}_i; \boldsymbol{\eta})\Big\}^{1-y_i} \quad ,
\end{aligned} \tag{2.8}$$

where $\boldsymbol{X} = \Big\{\boldsymbol{x}_i\Big\}_{i=1}^{n} \subset \mathbb{R}^m$ and $\boldsymbol{y} = \bigtimes_{i=1}^{n} \Big\{y_i\Big\} \in \Big\{0, 1\Big\}^{n}$.
So the core idea of maximum likelihood is to find the parameters which maximize $\mathcal{L}(\boldsymbol{\eta})$. Instead of maximizing the function in (2.8), it is more convenient to maximize the logarithm of the function:

$$l(\boldsymbol{\eta}) = \log \mathcal{L}(\boldsymbol{\eta}) = \log \mathbf{P}(\boldsymbol{y} \mid \boldsymbol{X}; \boldsymbol{\eta})$$

$$= \sum_{i=1}^{n} \Big\{y_i \cdot \log \mathbf{P}(y_i = 1 \mid \boldsymbol{x}_i; \boldsymbol{\eta}) + (1 - y_i) \cdot \log \mathbf{P}(y_i = 0 \mid \boldsymbol{x}_i; \boldsymbol{\eta})\Big\} \ .$$

If the above function is given probability distribution be convex, then the optimal solution can be determined as follows:

$$\frac{\partial l(\boldsymbol{\eta})}{\partial \eta_i} = 0 \qquad \forall i \quad 1 \leq i \leq p \ ,$$

where p is assigned to the number of parameters. In this regard, the common approach like Newton or quasi Newton can be used. In binary case, the log likelihood function is as follows:

$$
\begin{aligned}
l(\boldsymbol{\omega}, \omega_0) &= \log \mathbf{P}(\boldsymbol{\omega}, \omega_0) = \log \left(\prod_{i=1}^{n} \mathbf{P}(y_i \mid \boldsymbol{x}_i; \boldsymbol{\omega}, \omega_0) \right) \\
&= \sum_{i=1}^{n} \left\{ y_i \log \pi_l^{(i)} + \big(1 - y_i\big) \log \big(1 - \pi_l^{(i)}\big) \right\}, \qquad (2.9)
\end{aligned}
$$

where $\pi_l^{(i)}$ is referred to in the equation in (2.6), where the instance $\boldsymbol{x}_i$ is considered. In addition $\mathbf{P}(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\omega}, \omega_0) = \big(\pi_l^{(i)}\big)^{y^{(i)}}$. It is not difficult to check that the log likelihood function in (2.9), has a negative semidefinite Hessian matrix [15, 96]. Therefore, the function $-l(\boldsymbol{\omega}, \omega_0)$ it is a convex function and to find the optimal solution the usual methods like gradient descent or the Newton method can be employed. For the Newton method, the optimal solution is as follows:

$$
\boldsymbol{\omega}^{(n+1)} = \boldsymbol{\omega}^{(n)} - H^{-1} \nabla_{\boldsymbol{\omega}} \big(- l(\boldsymbol{\omega}) \big) \ ,
$$

where H and ∇ are referred to the Hessian matrix and gradient respectively.
Furthermore, in order to establish a monotone learner, considering the linear logistic regression as a base-line, the following optimization problem is taken into account:

$$
\max_{\boldsymbol{\omega}, \omega_0} \ \sum_{i=1}^{n} \left\{ y_i \log \pi_l^{(i)} + \big(1 - y_i\big) \log \big(1 - \pi_l^{(i)}\big) \right\} \qquad (2.10)
$$

s.t.

$$
\omega_i \geq 0 \qquad \forall i \in \{1, \ldots, m\} \qquad (2.11)
$$

The constraints in (2.11) indeed assure the monotonicity for the linear logistic regression. The objective function in (2.10) is still convex, although beside some constraints exist. To this end, Lagrange optimization can accomplish the optimization problem.

2.4.2 Margin Maximization Principle

In this section, the basic idea of a large margin approach, namely the linear margin, is presented. Assume some labeled instances, which are supposed to be *i.i.d.* given as follows:

$$
\mathcal{D} = \Big\{ (\boldsymbol{x}_i, y_i) \Big\}_{i=1}^{n} \subset \mathbb{R}^m \times \{-1, +1\} \ .
$$

The labels $\{-1, +1\}$, can be considered as negative and positive classes respectively. The labeled instances are separable by a hyperplane $\boldsymbol{\omega}^* \in \mathbb{R}^m$ and intercept $b \in \mathbb{R}$, if the following inequality is valid for all labeled instances:

$$\forall i \;\; 1 \leq i \leq n \qquad\qquad y_i \cdot \Big(\langle \boldsymbol{\omega}^*, \boldsymbol{x}_i \rangle + b \Big) \geq 1 \;\; . \tag{2.12}$$

In spite of this formulation, the instances with property $\langle \boldsymbol{\omega}^*, \boldsymbol{x}_i \rangle + b \geq 1$, belong to class $+1$, and the instances with property $\langle \boldsymbol{\omega}^*, \boldsymbol{x}_i \rangle + b \leq -1$ belong to class -1. Accounting for the fact that there are infinite hyperplanes, which can satisfy the above inequality, this reveals that there is a great deal of flexibility for existing solutions. To cope with this flexibility, which certainly contributes to the overfitting problem, Vapnik proposed to use the idea of SRM. The core idea is to find a trade-off between the goodness of generalization and complexity of model. To this end, he proposed the following risk:

$$\sum_{i=1}^{n} l\Big(y_i, \langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle \Big) + C||\boldsymbol{\omega}||^2$$

Here l is the loss function and C is the trade-off parameter. Since we assumed that the instances can be classified by a linear hyperplane, it can be concluded, a set of hyperplanes exist $\{\boldsymbol{\omega}_q\}_{q \in \mathcal{Q}}$, where for all $q \in \mathcal{Q}$, $\sum_{i=1}^{n} l(y_i, \langle \boldsymbol{\omega}_q, \boldsymbol{x}_i \rangle) = 0$. Therefore the problem boils down to minimizing the following term:

$$C||\boldsymbol{\omega}||^2$$

with additional constraints

$$y_i \cdot \Big(\langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle + b \Big) \geq 1 \;\; .$$

More formally the hyperplane can be determined as follows:

$$\begin{aligned} &\min_{\boldsymbol{\omega}, b} \frac{1}{2}||\boldsymbol{\omega}||^2 \\ &\text{s.t.} \\ &y_i \cdot \Big(\langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle + b \Big) \geq 1 \qquad\qquad \forall \; i \in \{1, \ldots, n\} \end{aligned}$$

This in essence is a quadratic programming optimization. In the light of above notations, the set of *support vectors* can be determined as follows:

$$\left\{ \boldsymbol{x}_{\mathcal{S}} \in \mathcal{D} \;\middle|\; |\langle \boldsymbol{\omega}^*, \boldsymbol{x}_{\mathcal{S}} \rangle + b| = 1 \right\} \;,$$

where $\boldsymbol{\omega}^*$ is the solution of the optimization problem above. The vector $\boldsymbol{\omega}^*$ is also called the *decision boundary*. In this case, the *decision boundary* is linear, however it can be quite non-linear. Then following, discuss in greater detail how to construct non-linear decision boundaries.

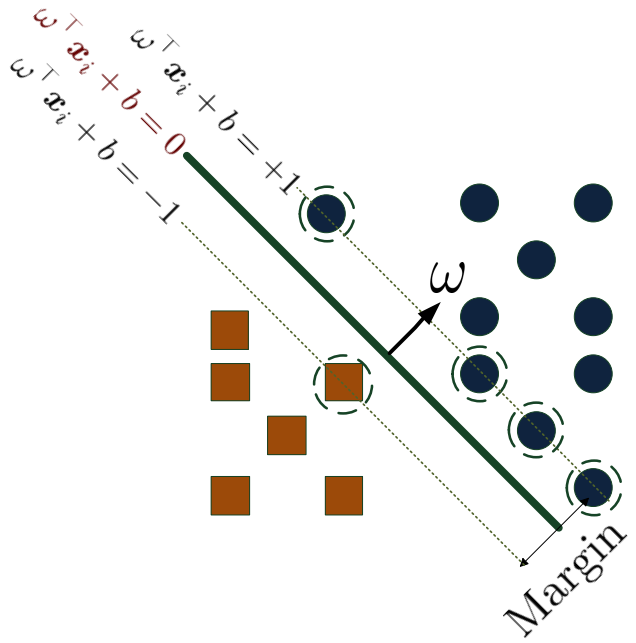

Figure 2.3: The illustration of separation of two classes by hyperplane $\langle \boldsymbol{\omega}, \boldsymbol{x} \rangle + b = 0$. The objects on boundary, which showed by dot circles, called the support vectors.

In Figure 2.3, the position of hyperplane is shown and as well the support vectors. Since the above equality holds for support vectors, the distance between support vectors and hyperplane can be computed as follows:

$$d(\boldsymbol{x}_{\mathcal{S}}, \mathbb{P}) = \frac{\boldsymbol{\omega}^\top \boldsymbol{x}_{\mathcal{S}} + b}{||\boldsymbol{\omega}||} = \frac{\pm 1}{||\boldsymbol{\omega}||} \;, \tag{2.13}$$

where $\mathbb{P} = \left\{ \boldsymbol{x} \in \mathbb{R}^m \;\middle|\; \langle \boldsymbol{\omega}, \boldsymbol{x} \rangle + b = 0 \right\}$. Therefore, the magnitude of margin is equal to $\frac{2}{||\boldsymbol{\omega}||}$, and hence the solution to the above constrained optimization problem,

is the hyperplane, which maximizes the distance between the instances of two different classes.

So far the assumption was, that the instances are linearly separable. One may also consider the case, in which the instances are not linearly separable, although the goal is to separate them by a linear hyperplane. This case is addressed as a soft margin. The core idea is to allow the classifier to make some mistakes, albeit as low as possible. In Figure 2.4 the idea is illustrated. This idea is again referred to as the structural risk minimization. To this end, assume the loss function is given. Then

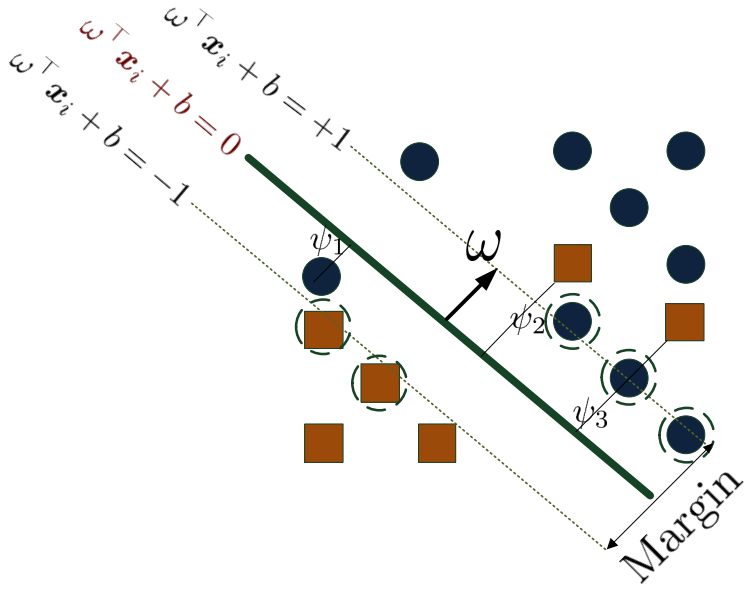

Figure 2.4: The illustration of separation of two classes by soft margin $\langle \boldsymbol{\omega}, \boldsymbol{x} \rangle + b = 0$. The objects on boundary, which showed by dot circles, called the support vectors.

the structural risk can be formulated as follows:

$$\sum_{i=1}^{n} l(y_i, \langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle) + C||\boldsymbol{\omega}||^2$$
$$= \sum_{i=1}^{n} \psi_i + C||\boldsymbol{\omega}||^2 \ .$$

The first term in the second equation, indeed, is considered for mistakes. Hence the linear soft margin can be formalized as follows:

$$\min_{\boldsymbol{\omega},\boldsymbol{\psi},b} \Big\{ \sum_{i=1}^{n} \psi_i + \frac{C}{2}||\boldsymbol{\omega}||^2 \Big\}$$

s.t.

$$y_i \cdot \Big(\langle \boldsymbol{\omega}^*, \boldsymbol{x}_i \rangle + b \Big) \geq 1 - \psi_i \qquad \forall\, i \in \{1, \ldots, n\}$$

$$\psi_i \geq 0 \qquad \forall\, i \in \{1, \ldots, n\}$$

This is again a constraint optimization problem, and the optimal solution can be found by quadratic programming optimization.

2.4.3 Kernel Methods

It is apparent that if the instances are not linearly separable, then linear SVM cannot solve the problem properly, i.e., some instances are miss-classified. In order to prevent miss-classification, the basic idea is to transfer the data (instances) to an upper space, of course with higher dimensionality, where the labeled instances can be separated linearly without any mistake. To this end, the core idea is to use kernels which can model the non-linear decision boundaries.

Before going into detail, we shall introduce some preliminaries. Assume the function $k(\cdot, \cdot)$ on domain $\mathcal{X} \times \mathcal{X}$ is defined as follows:

$$\begin{aligned} k &: \mathcal{X} \times \mathcal{X} \to \mathbb{R} \\ &(\boldsymbol{x}, \boldsymbol{x}') \mapsto k(\boldsymbol{x}, \boldsymbol{x}') \ . \end{aligned}$$

In order to establish the concept of the kernel, the following definitions and notations are needed as taken from "Kernel methods in Machine Learning" by Hofmann et al. in [57].

Definition 2.1 ***(Gram matrix)*** *Given a function $k(\cdot, \cdot)$ and inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \subset \mathbb{R}^m$, the $n \times n$ matrix*

$$K := \Big(k(\boldsymbol{x}_i, \boldsymbol{x}_j) \Big)_{ij}$$

is called Gram matrix of $k(\cdot, \cdot)$ with respect to $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.

Definition 2.2 ***(Positive definite matrix)*** *A real $n \times n$ symmetric matrix K_{ij} satisfying*

$$\sum_{i,j} h_i h_j K_{ij} \geq 0$$

for all $h_i \in \mathbb{R}$ is called positive definite. If equality only occurs for $h_1 = \ldots = h_n = 0$, then we shall call the matrix strictly positive definite.

Definition 2.3 ***(Positive definite kernel)*** *Let $\mathcal{X} \subset \mathbb{R}^m$ be a nonempty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which for all $n \in \mathbb{N}, \boldsymbol{x}_i \in \mathcal{X}, i \in \{1, \ldots, n\}$ gives rise to a positive definite Gram matrix is called a positive definite kernel. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which for all $n \in \mathbb{N}$ and distinct $\boldsymbol{x}_i \in \mathcal{X}$ gives rise to a strictly positive definite Gram matrix is called a strictly positive definite kernel.*

Theorem 2.2 ***(Mercer's Theorem)*** *A symmetric function $k(\cdot, \cdot)$ is a kernel if and only if for any finite sample $S = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ the Gram matrix for S is positive semidefinite.*

If for a given kernel $k(\cdot, \cdot)$, there is a mapping $\varphi : \mathbb{R}^m \to \mathbb{R}^p$ such that for all $\boldsymbol{x}, \boldsymbol{x}^* \in \mathbb{R}^m$, $k(\boldsymbol{x}, \boldsymbol{x}^*) = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}^*) \rangle$, the map $\varphi(\cdot)$ is called *feature mapping* with respect to the kernel $k(\cdot, \cdot)$.

Given n training examples $\left\{\boldsymbol{x}_i\right\}_{i=1}^n$ the optimal weights for a kernel is computed by the so-called *dual form* [107]. The setting in a large margin case is called the *primal form*. In fact from the *primal form*, the dual form can be derived [87]. So in this case, the goal is to find $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^n$ parameters, which minimize the following objective function under constraints:

$$\min_{\boldsymbol{\alpha}} \left\{ \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j \alpha_i \alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i=1}^n \alpha_i \right\}$$

s.t.

$$\sum_{i=1}^n y_i \alpha_i = 0$$

$$0 \leq \alpha_i \leq C \qquad \forall\, i,\ i \in \{1, \ldots, n\}$$

Here C is a typical SVM trade-off parameter. This problem actually can be solved by quadratic programming (QP), which given n instances has a computational complexity of $\mathcal{O}(n^3)$. Interestingly, from $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ and training examples, $\boldsymbol{\omega}$ can be computed as follows:

$$\boldsymbol{\omega} = \sum_{i=1}^{n} \alpha_i y_i \varphi(\boldsymbol{x}_i) \ ,$$

where $\varphi(\cdot)$ is the feature mapping corresponding to the kernel. In the following discussion, the kernels used in this thesis are introduced.

Polynomial Kernel

Let $\boldsymbol{x}, \boldsymbol{y}$ be two elements in $\mathbb{R}^m$. The polynomial kernel is defined as follows:

$$k(\boldsymbol{x}, \boldsymbol{y}) = \Big(\langle \boldsymbol{x}, \boldsymbol{y} \rangle + \lambda\Big)^d \ ,$$

where $d \in \mathbb{N}$ is the degree of polynomial kernel and also $\lambda \in \mathbb{R}$. This corresponds to the feature map $\varphi(\cdot)$ including all monomials $x_1^{i_1} \dots x_m^{i_m}$ where $i_j \in \mathbb{N}$, $\sum_{i=1}^{m} i_j = s$ and $0 \leq s \leq d$. For $d = 1$ it is called the linear kernel.

Gaussian Kernel (RBF)

Let $\boldsymbol{x}, \boldsymbol{y}$ be two elements in $\mathbb{R}^m$. The Gaussian kernel given parameter σ is defined as follows:

$$k_\sigma(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{y}||_2^2}{2\sigma^2}\right) \ ,$$

This kernel is also called the radial basis function (RBF).

2.5 Monotone Classifiers

The problem of monotone classification has received increasing attention in the machine learning community in recent years [7, 29, 35], despite having been introduced in the literature much earlier [11]. Meanwhile, several machine learning algorithms have been modified so as to guarantee monotonicity in attributes, including nearest neighbor classification [33], neural networks [91], decision tree learning [10, 85], rule induction [29], as well as methods based on isotonic regression [22] and piecewise linear models [28]. From a monotone learning point of view, specifically for classification purpose, the monotone classifiers are trained in a way that the learned classifiers satisfy monotonicity properties. In general, monotonicity means

that by increasing the magnitude of attributes jointly or separately, the corresponding class also increases or at least stays at the same level. More precisely, assume the classifier is given as follows:

$$CL : \mathcal{X}_1 \times \ldots \times \mathcal{X}_m \to \mathcal{Y} = \{-1, +1\} \ ,$$

where $\mathcal{X}_1 \times \ldots \times \mathcal{X}_m \subseteq \mathcal{X} \subset \mathbb{R}^m$. The classifier $CL(\cdot)$ is a monotone classifier, if

$$\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X} \quad s.t. \quad \boldsymbol{x}_i \preceq \boldsymbol{x}_j \quad \text{then} \quad CL(\boldsymbol{x}_i) \leq CL(\boldsymbol{x}_j) \ .$$

This definition can be extended to the ordinal classes, where there is a total order between classes. More concretely, assume the classes $\mathcal{Y} = \{y_1, \ldots, y_k\}$ with the following order are given:

$$y_1 \prec y_2 \prec \ldots \prec y_K \ .$$

Note that, $\{y_i\}_{i=1}^n$ are not necessarily real numbers. In that case the above definition can be extended as follows:

$$\text{If} \quad \boldsymbol{x}_i \preceq \boldsymbol{x}_j \quad \text{then} \quad CL(\boldsymbol{x}_i) \preceq CL(\boldsymbol{x}_j) \ .$$

It is worth mentioning that usually enforcing monotonicity to specific model can be accomplished in light of extra constraints [67, 98]. For instance, every linear SVM with positive parameters for $\boldsymbol{\omega}$ is a monotone classifier. Needless to say, it is not always possible to enforce the monotonicity to an arbitrary model.

3

The Choquet Integral as an Aggregation Function

In this chapter, the main ideas of *Multi Criteria Decision Making* (MCDM) and aggregation functions are presented. The aggregation functions are mostly used in the *Multi Criteria Decision Making* community. Loosely speaking, the task in multi criteria decision making is to select an object or an action between several alternatives. In this regard, each object or action can be characterized by its properties. Ultimately such properties can be aggregated, and the final decision (selection, ranking) is made. This chapter begins by giving the core ideas in multiple criteria decision making. Then the aggregation functions as powerful tools in MCDM are introduced in 3.1.3. Specifically the Choquet integral as an aggregation function in Section 3.4 is presented and its properties will be described. As it has been mentioned in introduction of thesis, the Choquet integral provides sound information in terms of interpretation. This issue is addressed as well in Section 3.6 in more the detail .

3.1 Multiple Criteria Decision Making

3.1.1 Introduction

To the best of our knowledge the initial idea of MCDM comes from philosophy, where the problem was to evaluate a premise by its *pro* and *contra* reasons [36]. In this regard, the pro (advantages) and contra (disadvantages) reasons were weighted, and finally the weights were compared. If the weights of advantages compared to the weights of disadvantages were larger, the premise would be accepted, otherwise it would be rejected. In fact, the issue was to select an alternative among of several alternatives, where every alternative can be assessed by its advantages and disadvantages. Generally, the selection of one alternative among all alternatives is not always a trivial task. The basic reason is, that it is quite rare that an alternative can cover all advantages, whereas it does not have any disadvantage. From this perspective, the task in MCDM is to rank the alternatives in a preferable way. This chapter starts by an simple example to give the main idea of MCDM:

Assume a company wants to employ a programmer. Since the company is operating in the U.S. and China, it needs a person, who is proficient in a foreign language. Also 70% of the projects are done using `Java`, 30% using `C+` and knowing `SQL` is preferable. Finally since the company faces complex problems, more educated employees are preferred. In addition, the company would not like to pay more than $60k$ \$ as salary to the employee. Accordingly every alternative can be described as a member of the following Cartesian product:

$$\Big\{\mathcal{P}(\{\texttt{Eng.},\texttt{Chin.}\})\Big\} \times \Big\{\mathcal{P}(\{\texttt{Java},\texttt{C+},\texttt{SQL}\})\Big\} \times \Big\{\texttt{B.Sc.},\texttt{M.Sc.},\texttt{PhD}\Big\} \times \Big\{\texttt{M},\texttt{L}\Big\},$$

where $\mathcal{P}$ is assigned to the powerset. Here `M` stands for more than $60k$ \$ and `L` stands for less than $60k$ \$ income. Note that here an empty set means the candidate cannot satisfy any preconditions regarding the specific condition. Needless to say, every alternative expects different amounts of income with respect to his/her qualifications. The task here is to find a programmer who can satisfy as much as possible the prerequisites and moreover expects an income less than $60k$ \$ as a salary.

3.1.2 General Idea

MCDM has been received considerable attention, especially from 50 years ago. This is indeed a subfield of operations research, which attempts to make a proper

choice among the given choices or to find the proper ranking/sorting among of the given choices. From an application point of view, the MCDM is used extensively in banking [6], transportation [27], urban management [113], energy and resource management [72], economy [71], energy planning [32] and financial management [23]. The task in MCDM can be summarized as choosing an alternative among given alternatives or sorting the alternatives. This task can be quite complex and confusing due to the fact that, every alternative can cover the benefits partially, i.e., usually an alternative cannot fully satisfy all expectations. Specifically in real applications, usually the criteria are conflicting, namely, increasing the satisfaction of certain criterion leads to decrease in the satisfaction of another criterion. In this regard, there is a need to make a choice in a transparent and consistent way. In general, every MCDM can be described by the following components:

- **Alternative:** An alternative is an object or an action which has a potential to be chosen.
- **Criteria:** In order to evaluate each alternative and to make them comparable, each alternative is characterized by some predefined criteria. Note that all alternatives are characterized by the same criteria.
- **Make a Decision:** Given the alternatives characterized by criteria, the decision is made by evaluating the alternatives and giving them scores, and ultimately by choosing an alternative, sorting the alternatives or ranking the alternatives (total or partial) among several alternatives.

Therefore, regarding the above definitions, given the set of criteria $C = \{c_1, \ldots, c_m\}$ each alternative can be evaluated as follows:

$$\boldsymbol{a} = (a_1, \ldots, a_m) \ ,$$

where $\forall i, 1 \leq i \leq m$, $a_i = f_i(\boldsymbol{a})$. The function $f_i(\cdot)$ is called the evaluation function w.r.t the criterion c_i, and indicates how high the alternative $\boldsymbol{a}$ can satisfy the criterion c_i. Commonly it is assumed that the range of the evaluation function is $[0, 1]$. 0 means the alternative cannot satisfy anything regarding criterion c_i and 1 means the alternative satisfies perfectly the criterion c_i. Hence,

$$\boldsymbol{a} = (a_1, \ldots, a_m) = \big(f_1(\boldsymbol{a}), \ldots, f_m(\boldsymbol{a})\big) \in [0, 1]^m$$

In order to make a choice or rank the alternatives, the existing methods in literature can be divided into three main sub-methods:

Outranking methods: The main idea in outranking methods is to define a preference relation (partial order) of alternatives in a pairwise manner. More precisely $A \prec B$, implies the alternative B is at least as good as the alternative A. To this end, the core idea is to consider the criteria which support the assertion ($A \prec B$) and the criteria which are against this. The initial idea has been proposed by Bernard Roy [86]. *Multiattribute utility:* The multiattribute methods usually consider the classical aggregation functions (we will describe the aggregation functions in the next section) and assign real numbers to each alternative. Therefore, they produce a total order given alternatives. *Non classical methods:* Finally the non-classical approaches mainly covered in the scholarly literature *decision rules* [55] and *fuzzy integrals* [50].
In this regard, while several well established methods such as ELECTRE (elimination and choice expressing reality) [37], PROMETHEE (preference ranking organization method for enrichment of evaluations) [18], TOPSIS (technique for order preference by similarity to ideal solution) [82, 112] and the MUTA (multi attribute utilities theory) [34] are counted as classical approaches, the aggregation functions underlying fuzzy measures are taken into consideration as a non-classical approach [42, 36]. In the following sections, these kind of aggregation functions will be described more in the details. Before continuing the topic, a new notation should be introduced.

So far the term attribute(s) was mentioned to characterize the feature space. Basically every instance is characterized by some given attributes. We shall make a distinction between *attribute* and *criterion*. From a MCDM point of view, criteria characterize the decision space. Every criterion is not a number, however, for each instance the evaluation of each criterion, namely, how good the alternative can satisfy specific criterion, is computed by its evaluation function (real value function). The term "criterion" is indeed often used in the decision making literature, where it suggests a monotone "the higher the better" influence. In the light of this definition, this fact comes down, that the higher the criterion, the higher the benefit is. Note that from now to the end of this thesis, the set of criteria are shown as follows:

$$C = \Big\{ c_1, \ldots, c_m \Big\} \ ,$$

where m is the number of criteria.

3.1.3 Aggregation Functions

As mentioned several times, one of the powerful tools from a decision making point of view is an aggregation function. The main duty of aggregation functions is to aggregate all advantages and disadvantages together and finally deliver them as a real (positive) number. Before going into details, we shall introduce the definition of aggregation functions.

Formally a function $A : [0,1]^m \to [0,1]$ is an aggregation function if it satisfies the following conditions [21, 51]:

- monotonicity: $\forall \boldsymbol{a}_i, \boldsymbol{a}_j \in [0,1]^m$, if $\boldsymbol{a}_i \prec \boldsymbol{a}_j$ then $A(\boldsymbol{a}_i) \leq A(\boldsymbol{a}_j)$
- Unanimity: $\forall \boldsymbol{a} = (a, \ldots, a) \in [0,1]^m, A(a, \ldots, a) = a$

The second condition immediately boils down to the following equalities:

$$A(0, \ldots, 0) = 0$$
$$A(1, \ldots, 1) = 1$$

An aggregation function is symmetry, if for any permutation σ on $\{1, \ldots, m\}$, $A(a_1, \ldots, a_m) = A(a_{\sigma(1)}, \ldots, a_{\sigma(m)})$.

In the following some important aggregation functions are presented:

Definition 3.1 ***(Weighted Mean)**: Given the weights $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_m)$, where $\omega_i \geq 0$ and $\sum_{i=1}^{m} \omega_i = 1$, the weighted mean operator is defined as follows:*

$$WM(a_1, \ldots, a_m) = \sum_{i=1}^{m} \omega_i a_i \ .$$

Specifically, when $\forall i,\ \omega_i = \frac{1}{m}$, it is called the arithmetic mean of dimension m.

Definition 3.2 ***(Ordered Weighted Averaging)****: Given the weights* $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_m)$*, where* $\omega_i \geq 0$ *and* $\sum_{i=1}^{m} \omega_i = 1$*, and moreover a decreasing permutation* π $(a_{\pi(i)} \geq a_{\pi(i+1)}$ *for a given* $\boldsymbol{a})$*, the OWA operator is defined as follows:*

$$OWA(a_1, \ldots, a_m) = \sum_{i=1}^{m} \omega_i a_{\pi(i)} \quad .$$

In the following sections, we give the preliminaries for introducing an important aggregation function underlying *fuzzy measure*.

3.2 The Choquet Integral as an Extension of Lebesgue Integral

Henri Lebesgue proposed the extension of the Riemann integral, where instead of considering the length of an interval as the weight, he presented the concept of measure to construct the integral. Such measures have a specific property, namely, countable additivity. More precisely, the measure of union of two distinctive sets is equal to the sum of measures of each set. Later on Gustave Choquet proposed the generalization of the Lebesgue integral in a way that he suggested corresponding measure can be non-additive measures. More formally, assume function $f : \mathcal{S} \to \mathbb{R}$ is measurable with respect to measure ν. The Choquet integral for function f respect to measure ν is defined as follows:

$$(c) \int_{\mathcal{S}} f \mathrm{d}\nu := \int_0^{\infty} \nu\Big(\{s \mid f(s) \geq x\}\Big) \mathrm{d}x + \int_{-\infty}^{0} \Big[\nu\Big(\{s \mid f(s) \geq x\}\Big) - \nu(\mathcal{S})\Big] \mathrm{d}x$$

It is worth mentioning that if ν is a σ-additive measure, the above definition comes down to the Lebesgue integral.

3.3 Fuzzy Measures

Before describing the details, some preliminaries about non-additive measures and especially fuzzy measure are introduced. As mentioned earlier, the crucial difference between the Choquet integral and the Lebesgue integral is the type of measure.

3.3.1 Non-Additive Measures

In this section the properties of non-additive measures are investigated in greater detail. Generally, non-additive measures do not satisfy the additive property. Hence, non-additive measures obviously are more flexible and can model a larger class of measures as well. In the following discussion, the main properties of such kind of measures are introduced.

Let $C = \{c_1, \ldots, c_m\}$ be a finite set of criteria and $\mu(\cdot)$ a measure $2^C \to [0, 1]$. For each $A \subseteq C$, we interpret $\mu(A)$ as the *weight* or, say, the *importance* of the set of elements A. As an illustration, one may think of C as a set of criteria (binary features) relevant for a job, like "speaking French" and "programming Java", and of $\mu(A)$ as the evaluation of a candidate satisfying criteria A (and not satisfying $X \setminus A$).

A standard assumption of a measure $\mu(\cdot)$, which is at the core of probability theory is additivity: $\mu(A \cup B) = \mu(A) + \mu(B)$ for all $A, B \subseteq X$ such that $A \cap B = \emptyset$. Unfortunately, additive measures cannot model any kind of interaction between elements: Extending a set of elements A by a set of elements B always increases the weight $\mu(A)$ by the weight $\mu(B)$, regardless of A and B. Suppose, for example, that the elements of two sets A and B are *complementary* in a certain sense. For instance, $A = \{\texttt{French}, \texttt{Spanish}\}$ and $B = \{\texttt{Java}\}$ could be seen as complementary, since both language skills and programming skills are important for the job. Formally, this can be expressed in terms of a positive interaction: $\mu(A \cup B) > \mu(A) + \mu(B)$. In the extreme case, when language skills and programming skills are indeed essential, $\mu(A \cup B)$ can be high although $\mu(A) = \mu(B) = 0$ (suggesting that a candidate lacking either language or programming skills is completely unacceptable). Likewise, elements can interact in a negative way: If two sets A and B are partly *redundant* or *competitive*, then $\mu(A \cup B) < \mu(A) + \mu(B)$. For example, $B = \{\texttt{Java}\}$ and $C = \{\texttt{C}, \texttt{C\#}\}$ might be seen as redundant, since one programming language does in principle suffice.

3.3.2 Fuzzy Measures and their Möbius Transforms

As mentioned before, non-additive measures are characterized by different properties. Among all non-additive measures, there is a specific kind of measure called *fuzzy measures*. They have a unique property called monotonicity. This property guarantees, that the measure of each subsets of a set $\mathcal{S}$ have the magnitude smaller than (or equal to) the measure of $\mathcal{S} \subset C$.

Definition 3.3 (Fuzzy measure) *Let $C = \{c_1, c_2, \ldots, c_m\}$ be a finite set. A discrete fuzzy measure (also called capacity) is a set function $\mu : 2^C \rightarrow [0, 1]$ which is monotonic ($\mu(A) \leq \mu(B)$ for $A \subseteq B \subseteq C$) and normalized ($\mu(\emptyset) = 0$ and $\mu(C) = 1$). A fuzzy measure μ is called additive if $\mu(A \cup B) = \mu(A) + \mu(B)$ for all $A, B \subset C$ such that $A \cap B = \emptyset$. Obviously, in the case of an additive measure, $\mu(A)$ is simply obtained as follows [95]:*

$$\mu(A) = \sum_{i \in A} \mu(\{i\}) \tag{3.1}$$

Definition 3.4 (Möbious transform) *The Möbius transform $\boldsymbol{m}_\mu$ of a fuzzy measure μ is defined as follows:*

$$\boldsymbol{m}_\mu(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|} \mu(B)$$

for all $A \subseteq C$.

A useful property of the Möbius transform, that we shall exploit later on for learning the Choquet integral, it allows for reconstructing the underlying fuzzy measure:

$$\mu(B) = \sum_{A \subseteq B} \boldsymbol{m}_\mu(A)$$

for all $B \subseteq C$.

3.3.3 Monotonicity Constraints

Given an arbitrary measure $\mu : \{c_1, \ldots, c_m\} \rightarrow [0, 1]$, with additional assumption $\mu(\{c_1, \ldots, c_m\}) = 1$, one natural question is, whether the measure $\mu(\cdot)$ is a fuzzy measure or not. In order to check this issue, the basic idea is to check the following inequalities:

$$\mu(L) \leq \mu(K) \quad \forall L, K \;\; L \subset K \subseteq \{c_1, \ldots, c_m\} \;\; .$$

Since each subset of $\{c_1, \ldots, c_m\}$ is compared with all its subsets $3^m - 2^m$ constraints must be checked,

$$\sum_{i=1}^{m} \binom{m}{i} (2^i - 1) = \sum_{i=1}^{m} \binom{m}{i} 2^i - \sum_{i=1}^{m} \binom{m}{i} = 3^m - 2^m \;\; .$$

Fortunately, the last two constraints can be represented in a more compact way, exploiting a transitivity property:

$$\mu(L) \leq \mu(K) \qquad \forall L, K \;\; L \subset K \subseteq \{c_1, \ldots, c_m\}, \; |K| = |L| + 1 \;\; .$$

Respectively in terms of Möbius transform the above constraints can be reformulated as follows:

$$\sum_{B \subseteq A \setminus \{c_i\}} \boldsymbol{m}\Big(B \cup \{c_i\}\Big) \geq 0 \qquad \forall A \subseteq C, c_i \in C$$

This representation reduces the number of constraints to $m2^{m-1}$, which, despite still being large, is a significant reduction in comparison with the original formulation.

3.3.4 k-additivity

There is a close connection between a fuzzy measure and the Möbius transform; given a fuzzy measure the so-called Möbius transform can be constructed conversely as well. From a computational point of view, there is a complexity reduction, so that instead of considering all values for the Möbius transform only a subset of values are taken into account. This reduction is called k-additivity, where k is referred to the level-complexity.

Definition 3.5 (k-Additivity) *A fuzzy measure μ is said to be k-order additive or simply k-additive if k is the smallest integer such that $\boldsymbol{m}(A) = 0$ for all $A \subseteq C$ with $|A| > k$.*

Thus, while a Choquet integral is determined by 2^m coefficients in general, the k-additivity of the underlying measure reduces the number of required coefficients to at most

$$\sum_{i=1}^{k} \binom{m}{i} \;\; .$$

3.4 The Discrete Choquet Integral

In previous section, the general idea of the Choquet integral was introduced. In this section we restrict ourselves to discrete cases, where the measure ν acts solely on finite domain in terms of cardinality.

So far, the criteria c_i were simply considered as binary features, which are either present or absent. Mathematically, $\mu(A)$ can thus also be seen as an *integral* of the indicator function of A, namely the function f_A given by $f_A(c) = 1$ if $c \in A$ and $= 0$ otherwise. Now, suppose that $f : C \to \mathbb{R}_+$ is any non-negative function that assigns a *value* to each criterion c_i; for example, $f(c_i)$ might be the degree to which a candidate satisfies criterion c_i. An important question, then, is how to *aggregate* the evaluations of individual criteria, i.e., the values $f(c_i)$, into an overall evaluation, in which the criteria are properly weighted according to the measure μ. Mathematically, this overall evaluation can be considered as an integral $\mathcal{C}_\mu(f)$ of the function f with respect to the measure μ.

Indeed, if μ is an additive measure, the standard integral just corresponds to the *weighted mean*

$$\mathcal{C}_\mu(f) = \sum_{i=1}^{m} w_i \cdot f(c_i) = \sum_{i=1}^{m} \mu(\{c_i\}) \cdot f(c_i) \ , \tag{3.2}$$

which is a natural aggregation operator in this case. A non-trivial question, however, is how to generalize (3.2) in the case when μ is non-additive.

This question, namely how to define the integral of a function with respect to a non-additive measure (not necessarily restricted to the discrete case), is answered in a satisfactory way by the Choquet integral, which has first been proposed for additive measures by [109] and later on for non-additive measures by [24]. The point of departure of the Choquet integral is an alternative representation of the "area" under the function f, which, in the additive case, is a natural interpretation of the integral. Roughly speaking, this representation decomposes the area in a "horizontal" instead of a "vertical" manner, thereby making it amenable to a straightforward extension of the non-additive case. More specifically, note that the weighted mean can be expressed as follows:

$$\begin{aligned}\sum_{i=1}^{m} f(c_i) \cdot \mu(\{c_i\}) &= \sum_{i=1}^{m} \Big(f(c_{(i)}) - f(c_{(i-1)})\Big)\Big(\mu(\{c_{(i)}\}) + \ldots + \mu(\{c_{(m)}\})\Big) \\ &= \sum_{i=1}^{m} \Big(f(c_{(i)}) - f(c_{(i-1)})\Big) \cdot \mu\Big(A_{(i)}\Big) \ ,\end{aligned}$$

where $(\cdot)$ is a permutation of $\{1, \ldots, m\}$ such that $0 \leq f(c_{(1)}) \leq f(c_{(2)}) \leq \ldots \leq f(c_{(m)})$ (and $f(c_{(0)}) = 0$ by definition), and $A_{(i)} = \{c_{(i)}, \ldots, c_{(m)}\}$; see Figure 3.1 as an illustration.

Now, the key difference between the left and right-hand side of the above expression is that, whereas the measure μ is only evaluated on single elements c_i on

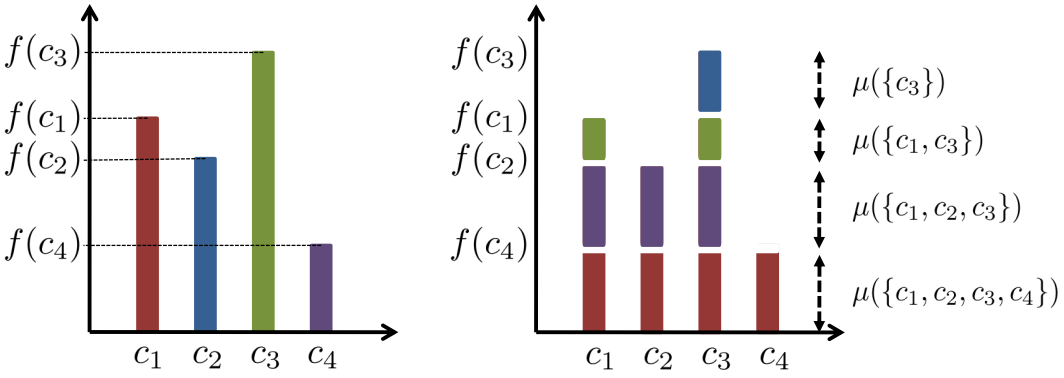

Figure 3.1: Vertical (left) versus horizontal (right) integration. In the first case, the height of a single bar, $f(c_i)$, is multiplied with its "width" (the weight $\mu(\{c_i\})$), and these products are added. In the second case, the height of each horizontal section, $f(c_{(i)}) - f(c_{(i-1)})$, is multiplied with the corresponding "width" $\mu(A_{(i)})$.

the left, it is evaluated on *subsets* of elements on the right. Thus, the right-hand side suggests an immediate extension to the case of non-additive measures, namely the Choquet integral, which, in the discrete case, is formally defined as follows:

$$\mathcal{C}_\mu(f) = \sum_{i=1}^{m} \big(f(c_{(i)}) - f(c_{(i-1)})\big) \cdot \mu(A_{(i)})$$

In terms of the Möbius transform of μ, the Choquet integral can also be expressed as follows:

$$\begin{aligned}
\mathcal{C}_\mu(f) &= \sum_{i=1}^{m} \big(f(c_{(i)}) - f(c_{(i-1)})\big) \cdot \mu(A_{(i)}) \\
&= \sum_{i=1}^{m} f(c_{(i)}) \cdot (\mu(A_{(i)}) - \mu(A_{(i+1)})) \\
&= \sum_{i=1}^{m} f(c_{(i)}) \sum_{R \subseteq T_{(i)}} \boldsymbol{m}(R) \\
&= \sum_{T \subseteq C} \boldsymbol{m}(T) \times \min_{i \in T} f(c_i) \qquad (3.3)
\end{aligned}$$

where $T_{(i)} = \big\{S \cup \{c_{(i)}\} \,|\, S \subset \{c_{(i+1)}, \ldots, c_{(m)}\}\big\}$.

3.5 An Application of the Choquet Integral in MCDM

Back to the last question about hiring an employee, let us assume the applications of some candidates are given. Moreover, suppose the task is to find the optimal candidate and also rank the candidates based on their qualifications. To this end, the Choquet integral can be taken into consideration. As mentioned, the space of criteria can be described as follows:

$$\Big\{\mathcal{P}(\{\texttt{Eng.}, \texttt{Chin.}\})\Big\} \times \Big\{\mathcal{P}(\{\texttt{Java}, \texttt{C+}, \texttt{SQL}\})\Big\} \times \Big\{\texttt{B.Sc.}, \texttt{M.Sc.}, \texttt{PhD}\Big\} \times \Big\{\texttt{M}, \texttt{L}\Big\}$$

Let us call the criteria as follows:

c_1 : Eng.	c_3 : Java	c_6 : B.Sc.	c_9 : L
c_2 : Chin.	c_4 : c+	c_7 : M.Sc.	
	c_5 : SQL	c_8 : PhD	

Then the set of criteria is $C = \{c_1, \ldots, c_9\}$. Note that the criterion L is a dichotomous criterion $\{0, 1\}$, therefore, the criterion M can be defined in the absence of criterion L. So given the qualification of the applicant, the goal is to find an evaluation. Assume three applicants with the following qualifications already applied:

A_1	$Eng.(.8)$	$Java(.9)$	$B.Sc.(.7)$			$L(1)$
A_2	$Eng.(.5)$	$Java(.7)$	$B.Sc.(.4)$	$M.Sc.(.6)$		$L(1)$
A_3	$Eng.(.5)$	$Java(.3)$	$B.Sc.(.6)$	$M.Sc.(.7)$	$PhD(.8)$	$L(0)$

The number in parentheses are assigned to the degree that the applicant can satisfy a specific criterion. The degree in this case ranges from $[0, 1]$. As can be seen, the first applicant can cover only 4 criteria, although, by a high degree. The next applicant can cover more criteria, however the degrees of satisfaction are not high. Obviously the third applicant also covers 5 out 9 criteria, but at the cost of lower degrees, and he demands an income more than $60k$. As is clear, finding the optimal applicant, even for this small set, is not an easy task. In order to aggregate all qualifications together and quantify a score to represent these qualifications, the Choquet integral as a powerful aggregation function can be used. Before employing the Choquet integral, the fuzzy measure must be given or determined by some experts. In addition

suppose the weight of each criterion and the joint weights of criteria are given as follows in addition:

$$\mu\big(\{c_l,\ldots,c_m\}\big) := \Big(\frac{\big|\{c_l,\ldots,c_m\}\big|}{9}\Big)^2 \ .$$

Since $\mu(\emptyset)=0$, $\mu(C)=1$ and $\mu(A)<\mu(B)$, when $A\subset B$, hence, essentially $\mu(\cdot)$ is a fuzzy measure. Therefore it is possible to apply the Choquat integral underlying $\mu(\cdot)$. The following scores are associated with the qualifications of each candidate computed by the Choquet integral.

$$\begin{aligned}
\mathcal{C}_\mu(A_1) &= .7\times\frac{16}{81}+.1\times\frac{9}{81}+.1\times\frac{4}{81}+.1\times\frac{1}{81}=.1556\\
\mathcal{C}_\mu(A_2) &= .4\times\frac{25}{81}+.1\times\frac{16}{81}+.1\times\frac{9}{81}+.1\times\frac{4}{81}+.3\times\frac{1}{81}=.1630\\
\mathcal{C}_\mu(A_3) &= .3\times\frac{25}{81}+.2\times\frac{16}{81}+.1\times\frac{9}{81}+.1\times\frac{4}{81}+.1\times\frac{1}{81}=.1494
\end{aligned}$$

So as can be seen, the second applicant received an obviously better score. With respect to the scores, the candidates are ranked as follows:

$$A_2 \succ A_1 \succ A_3$$

Note that, this is only an example to show, how the evaluation procedure by the Choquet integral can be carried out. What are more desirable in real applications are the values of the fuzzy measure. We assumed here, the values are given, however from an application point of view, it is more expected to estimate the values from some observations. To this end, in Chapters 4, 5 and 6 these issues are addressed in more detail.

Commensurability

In the last example, although the criteria are not comparable, there is a gradation possible that shows how well a candidate can satisfy a specific criterion. These degrees are shown in parentheses. In general, the criteria are not comparable, e.g., foreign language and income. In order to make the criteria comparable, it is necessary to use same scale for all of them. To this end, instead of criteria, the degree of satisfaction, that means, how good an instance can satisfy a specific criterion, should be taken into account. In order to have a consistent scale, this degree should range in interval $[a,b]$. Specifically the Choquet integral should be in $[0,1]$. The

simple reason is, the Choquet integral as an aggregation function has domain on $[0,1]^m$. The extreme cases, namely, 0 means the instance satisfies nothing, and 1 means the instance satisfies perfectly the criterion. In practice, it is common to use a utility (aggregation) function $\mathcal{U} : D(c_i) \to [0,1]$, where $D(c_i)$ is referred to domain of criterion c_i. This utility function is usually determined by an expert.

3.6 Interpretability of the Choquet Integral

One of the key features of the Choquet integral is interpretability. In particular, the Choquet integral (or, more specifically, the underlying fuzzy measure) provides natural measures of the importance of individual attributes and the interaction between pairs (or even groups) of attributes. Such measures are not only useful to understand the model, but can also be seen as a kind of feature selection process.

3.6.1 Shapley Index

From an interpretation point of view, a natural question is how high is the influence of criteria c_i. In the linear case, the answer is pretty simple. The influence of each criterion corresponds to the strength of the corresponding coefficient. In addition, the sign of each coefficient shows in which direction they are effective. More precisely, given the linear function $f(\cdot)$ as follows:

$$f(\boldsymbol{x}) = \omega_1 x_1 + \ldots + \omega_m x_m$$

the magnitude of each coefficient, namely $|\omega_i|$ demonstrate the influence of criterion c_i. Also $sign(\omega_i)$ corresponds to effective direction.
Seen from this point of view, measuring the importance of a criterion c_i becomes obviously more involved if μ is non-additive. In the literature, measures of this kind have been proposed. Lloyd Shapley proposed in 1953 the so-called Shapley value [90]. From a game theory point of view, a Shapley value computes the distribution to each cooperative game. Given a fuzzy measure μ on C, the *Shaply value* (or importance index) of c_i is defined as follows:

$$\varphi(c_i) = \sum_{A \subseteq C \setminus \{c_i\}} \frac{1}{m\binom{m-1}{|A|}} \Big(\mu(A \cup \{c_i\}) - \mu(A)\Big)$$

The Shaply value of μ is the vector $\boldsymbol{\varphi}(\mu) = \Big(\varphi(c_1), \ldots, \varphi(c_m)\Big)$. One can show that $0 \leq \varphi(c_i) \leq 1$ and $\sum_{i=1}^{m} \varphi(c_i) = 1$. Thus, $\varphi(c_i)$ is a measure of the *relative*

importance of c_i. Obviously, $\varphi(c_i) = \mu(\{c_i\})$ if μ is additive.

The Concept of Importance of Criteria (Shapley Index)

The basic idea of importancy is to assign a real positive value to each criteria. This value can be seen as a weight or let us say importancy for each criterion. The fuzzy measure itself does not describe anything about importancy, although from an interpretation and an application point of view, it is a crucial question how important the criterion $\{c_i\}$ is. To this end, the basic idea which originally comes from game theory, is to add criteria $\{c_i\}$ to other existing criteria, and check how the score has changed. More precisely, the following values can demonstrate such change:

$$\delta_i^A(\mu) := \mu(A \cup \{c_i\}) - \mu(A) \quad \forall A \subseteq C \setminus \{c_i\}$$

Additionally, since $\delta_i^A(\mu)$ is defined for each subset of $C \setminus \{c_i\}$, and all of them should be taken into consideration, the idea is taking on an average, which also is in agreement with weighted arithmetic mean. Moreover, since these values should indicate the importance of each criterion, they necessarily should be comparable. Therefore, for averaging of these values a normalization factor is needed, which guarantees all values ranging between two given bounds. Since for coalition $A \subseteq C \setminus \{c_i\}$, there are $\binom{m-1}{|A|}$ subsets, which have cardinality $|A|$, therefore for normalization $\binom{m-1}{|A|}$ factor is assumed. More precisely, Lloyd Stowell Shapley (1953) proposed an index for each criterion. Given a fuzzy measure μ on C, the *Shaply value* (or importance index) of c_i is defined as follows:

$$\varphi(c_i) = \sum_{A \subseteq C \setminus \{c_i\}} \frac{1}{m\binom{m-1}{|A|}} \Big(\mu(A \cup \{c_i\}) - \mu(A)\Big)$$

3.6.2 Interaction Index

The *interaction index* between criteria c_i and c_j, as proposed by Murofushi and Soneda [78], is defined as follows:

$$I(c_i, c_j) = \sum_{A \subseteq C \setminus \{c_i, c_j\}} \frac{\mu(A \cup \{c_i, c_j\}) - \mu(A \cup \{c_i\}) - \mu(A \cup \{c_j\}) + \mu(A)}{(m-1)\binom{m-2}{|A|}} .$$

This index ranges between -1 and 1 and indicates a positive (negative) interaction between criteria c_i and c_j if $I_{i,j} > 0$ ($I_{i,j} < 0$).

The interaction index can also be expressed in terms of the Möbius transform:

$$I(c_i, c_j) = \sum_{K \subseteq C \setminus \{c_i, c_j\}, |K|=k} \frac{1}{k+1} \boldsymbol{m}\Big(\{c_i, c_j\} \cup K\Big) .$$

Furthermore, as proposed by Grabisch [46], the definition of interaction can be extended to more than two criteria, i.e., to subsets $T \subseteq X$:

$$I(T) = \sum_{k=0}^{m-|T|} \frac{1}{k+1} \sum_{K \subseteq C \setminus T, |K|=k} \boldsymbol{m}\Big(T \cup K\Big) .$$

Interestingly, the Shaply value for a 2-additive case can also be expressed in terms of the interaction index [46]:

$$\varphi(c_i) = \boldsymbol{m}(\{c_i\}) + \frac{1}{2} \sum_{c_j \in C \setminus \{c_i\}} I(c_i, c_j) .$$

The Concept of Interaction (Interaction Index)

The so-called *Shapley index* quantifies the weight of an individual criterion. However from an interpretational point of view, quantifying the weight of two criteria jointly is also useful. To quantify this weight, the main idea is the same as the *Shapley index*. Let us define the $\delta_i^A(\mu),\ \delta_j^A(\mu),\ \delta_{i,j}^A(\mu)$ as follows:

$$\begin{aligned}
\delta_i^A(\mu) &:= \mu(A \cup \{c_i\}) - \mu(A) && \forall A \subseteq C \setminus \{c_i\} \\
\delta_j^A(\mu) &:= \mu(A \cup \{c_j\}) - \mu(A) && \forall A \subseteq C \setminus \{c_j\} \\
\delta_{i,j}^A(\mu) &:= \mu(A \cup \{c_i, c_j\}) - \mu(A) && \forall A \subseteq C \setminus \{c_i, c_j\}
\end{aligned}$$

Here the idea is to compare the value of a fuzzy measure after adding criteria $\{c_i, c_j\}$ to coalition A with the value of a fuzzy measure after adding criterion $\{c_i\}$ to coalition A and as well adding criterion $\{c_j\}$ to coalition A. More formally for every

coalition of criteria A, the following value can measure such change:

$$\begin{aligned}&\delta^A_{i,j}(\mu) - \delta^A_i(\mu) - \delta^A_j(\mu)\\ &= \mu(A \cup \{c_i, c_j\}) - \mu(A \cup \{c_i\}) - \mu(A \cup \{c_j\}) + \mu(A)\end{aligned}$$

In general, since there are several coalitions, the idea is to consider the averaged value of all coalitions. To this end, for each coalition, the normalization value can be computed as the number of all subsets of $C \setminus \{c_i, c_j\}$ which has cardinality $|A|$ (A is coalition), namely

$$\frac{1}{\binom{m-2}{|A|}}\,.$$

The above value refers to one coalition; A, however in total there are $m-1$ coalitions. Therefore, for a normalization factor we should take into account all of them, namely $(m-1)^{-1}$ factor. This normalization is also completely in agreement with weighted arithmetic mean. More formally: The *interaction index* between criteria c_i and c_j, as proposed by Murofushi and Soneda [78], is defined as follows:

$$I(c_i, c_j) = \sum_{A \subseteq C \setminus \{c_i, c_j\}} \vartheta_A \cdot \Big(\mu(A \cup \{c_i, c_j\}) - \mu(A \cup \{c_i\}) - \mu(A \cup \{c_j\}) + \mu(A)\Big)$$

with

$$\vartheta_A = \frac{1}{\binom{m-2}{|A|}}\,.$$

4

Monotone Learning by Using the Choquet Integral - Maximum Likelihood Approach

This chapter is devoted mainly for monotone classifiers. As mentioned earlier in 2.5, the duty of monotone classifier is to assure monotonicity. In general, the classifiers cannot assure monotonicity properties. Usually the ensuring the monotonicity for a classifier can be done in light of auxiliary constraints. Such constraints restrict the hypothesis space into a sub hypothesis space, where all candidates (hypotheses) are monotone classifiers. So in the case of the Choquet integral, our model satisfies monotonicity properties. Seen from this view, the hypothesis space is already restricted to the monotone functions, and each of which can satisfy monotonicity properties. However, from a learning point of view, one natural question is, which of them should be taken into account as a proper monotone classifier given some observations. To this end, the maximum likelihood estimation can provide the optimal solution. The general idea of maximum likelihood principle was given in Section 2.3. Actually the main idea is to define the posterior probability using the Choquet integral. Then immediately it is possible to employ maximum likelihood principle. In this regard, this chapter presents a framework to learn a family of monotone classifier underlying the Choquet integral with means of maximum likelihood esti-

mation. This chapter starts by describing our approaches to tackle the problem of ordinal classification in binary case. In this regard, the so-called *choquistic regression* is presented. Then in Section 4.2 the approaches for monotone ordinal classifiers are described. Specifically in Subsection 4.2.3 the *ordinal choquistic regression* as a generalization of ordinal logistic regression is proposed. Finally, in Section 4.3 the related works and researches basically respect to the Choquet integral are presented. Parts of this chapter were already published in [97, 98, 99, 100, 101]

4.1 Algorithms for Learning Monotone Binary Classifiers

One of the classical problem from a machine learning perspective is classifying the instances, which are defined by some attributes and labeled by two different labels. This problem is addressed as binary class classification in Section 2.2.3. Basically, the goal is to predict the class of a new query while some samples were already seen. For instance, assume some articles in the field of mathematics are given. Moreover assume each article is characterized by some criteria, e.g., impact factor and immediacy index and the output is whether the article is accepted or rejected. The goal here is, to predict the status (accept/reject) for a new article, characterized by given criteria.
As mentioned, the goal is to learn a monotone classifier. By the definition, a classifier $CL(\cdot)$ is a monotone classifier, if

$$\boldsymbol{x} \preceq_P \boldsymbol{x}^*, \quad \text{then} \quad CL(\boldsymbol{x}) \preceq CL(\boldsymbol{x}^*)$$

Here $CL(\cdot)$ is referred to as the output of classifier $CL(\cdot)$ given an observation. What immediately follows down from the definition is, that the classes are comparable and hence there is an order between two classes. Loosely speaking, in binary case the assumption is class 1 (positive) is better than class 0 (negative).

4.1.1 Linear Logistic Regression

The basic idea of linear logistic regression is to learn a dependency between input variables and their responses. More precisely, linear logistic regression modifies linear regression for the purpose of predicting (probabilities of) discrete classes instead of real-valued responses. To this end, the posterior probability of the positive class (and hence of the negative class) is modeled as a linear function of the input

attributes. Since the model has a probabilistic structure, it can be interpreted in a probabilistic way. In the conventional linear regression, one of the key assumptions is that errors of prediction $y - \hat{y}$ are normally distributed. When y only takes the values 0 and 1, this assumption is impossible to justify.

As mentioned in Section 2.4, the idea of linear logistic regression is to model the dependency between input variables and their responses. For this purpose, linear logistic regression takes advantages of the sigmoid function. More specifically, since a linear function does not necessarily produce values in the unit interval, the response is defined as a generalized linear model, namely in terms of the logarithm of the probability ratio:

$$\log\left(\frac{\mathbf{P}(y=1\,|\,\boldsymbol{x})}{\mathbf{P}(y=0\,|\,\boldsymbol{x})}\right) = \omega_0 + \boldsymbol{\omega}^\top \boldsymbol{x} \ , \tag{4.1}$$

where $\boldsymbol{w} = (w_1, \ldots, w_m)^\top \in \mathbb{R}^m$. A positive regression coefficient $\omega_i > 0$ means that an increase of the predictor variable x_i will increase the probability of a positive response, while a negative coefficient implies a decrease of this probability. Besides, the larger the absolute value $|\omega_i|$ of the regression coefficient, the stronger the influence of x_i. Since $\mathbf{P}(y = 0\,|\,\boldsymbol{x}) = 1 - \mathbf{P}(y = 1\,|\,\boldsymbol{x})$, a simple calculation leads to the posterior probability

$$\pi_l := \mathbf{P}(y=1\,|\,\boldsymbol{x}) = \frac{1}{1+\exp(-\omega_0 - \boldsymbol{\omega}^\top \boldsymbol{x})} \ . \tag{4.2}$$

The linear logistic regression also has several advantages. It is interpretable, i.e., the strength of each coefficient indicates how important the corresponding attribute can be. Also the direction of each coefficient $(-/+)$ can indicate in which direction the attribute has direct influence. In addition, it is easy to assure monotonicity for a linear logistic regression by enforcing positive coefficients. These advantages make linear logistic regressions more desirable from an application point of view.

Although the linearity of the above model is a strong restriction from a learning point of view, the possibility of interactions between predictor variables has of course also been noticed in the statistical literature [65]. A standard way to handle such interaction effects is to add interaction terms to the linear function of predictor variables. The simplest type of dependency is a linear relationship:

$$y = \sum_{i=1}^{m} \alpha_i x_i + \epsilon \ , \tag{4.3}$$

where $\alpha_1, \ldots, \alpha_m$ are real coefficients and ϵ is an error term. Monotonicity can be guaranteed quite easily for (4.3), since monotonicity in x_i is equivalent to the constraint $\alpha_i \geq 0$. Another important advantage of (4.3) as mentioned already, is its comprehensibility. In particular, the direction and strength of influence of each predictor x_i are directly reflected by the corresponding coefficient α_i.

Perhaps the sole disadvantage of a linear model is its inflexibility and, along with this, the supposed absence of any *interaction* between the variables: The effect of an increase of x_i is always the same, namely $\partial y / \partial x_i = \alpha_i$, regardless of the values of all other attributes. In many real applications, this assumption is not tenable. Instead, more complex, non-linear models are needed to properly capture the dependencies between the inputs x_i and the output y.

Increased flexibility, however, typically comes at the price of a loss in terms of the two previous criteria: comprehensibility is hampered, and monotonicity is more difficult to assure. In fact, as soon as an interaction between attributes is allowed, the influence of an increase in x_i may depend on all other variables, too. As a simple example, consider the extension of (4.3) by the addition of *interaction terms*, a model which is often used in statistics:

$$Y = \sum_{i=1}^{m} \alpha_i x_i + \sum_{1 \leq i < j \leq m} \alpha_{ij} x_i x_j + \epsilon \; . \tag{4.4}$$

For this model, $\partial y / \partial x_i$ is given by $\alpha_i + \sum_{j \neq i} \alpha_{ij} x_j$ and depends on the values of *all* other attributes, which means that, depending on the context as specified by these values, the monotonicity condition may change from one case to another. Consequently, it is difficult to find simple *global* constraints on the coefficients that assure monotonicity. For example, assuming that all attributes are non-negative, it is clear that $\alpha_i \geq 0$ and $\alpha_{ij} \geq 0$ for all $1 \leq i \leq j \leq m$ will imply monotonicity. While being sufficient, however, these constraints are non-necessary conditions, and may therefore impose restrictions on the model space that are more far-ranging than desired; besides, negative interactions cannot be modeled in this way. Quite similar problems occur for commonly used non-linear methods in machine learning, such as neural networks and kernel machines. Seen from this view, a linear logistic regression detects dependencies poorly. Actually, it considers the relationship between attributes and responses indeed quite independent.

Maximum Likelihood Estimation

In order to find the optimal generalization given some observations, one possibility is to employ the maximum likelihood estimation. Assume some observations with

corresponding responses are given, where the observations are assumed independent identically distributed:

$$\mathcal{D} = \Big\{(\boldsymbol{x}_i, y_i)\Big\}_{i=1}^{n} \subset \mathbb{R}^m \times \Big\{0, 1\Big\} \ .$$

The main idea of the maximum likelihood estimation was introduced in Subsection 2.3.1. In this section, it is used for a specific case, namely, binary classification. To this end, the optimal parameters can be found by the approach presented in Section 2.4.

4.1.2 Choquistic Regression

The linear logistic regression has several advantages, e.g., it is interpretable and comprehensible. It is extensively used in many scientific fields, like economics, psychology and sociology. Moreover it is easy to enforce monotonicity to linear logistic regression model by enforcing positive coefficients. But since the base line model is linear, it is not possible to model any dependency between attributes and response. Also it is not possible to model non-linear decision boundaries by using a linear model. Those disadvantages are core motivations to propose the extended model (extension) for linear logistic regression. However one non-trivial question is, how is it possible to extend the linear logistic model while preserving these advantages.
In order to model non-linear dependencies between predictor variables and response, and to take interactions between predictors into account, it is proposed to extend the logistic regression model by replacing the linear function by the Choquet integral.

$$\pi_c := \mathbf{P}(y = 1 \,|\, \boldsymbol{x}) = \frac{1}{1 + \exp\Big(-\gamma\big(\mathcal{C}_\mu(f_x) - \beta\big)\Big)} \ , \tag{4.5}$$

where $\mathcal{C}_\mu(f_x)$ is the Choquet integral (with respect to the measure μ) of the evaluation function $f_x : \{c_1, \cdots, c_m\} \to [0, 1]$ that maps each attribute c_i to a value $x_i = f_x(c_i)$; $\beta, \gamma \in \mathbb{R}$ are constants.

From Linear Logistic Regression to Choquistic Regression

In order to see that our model (4.5) is a proper generalization of standard logistic regression, recall that the Choquet integral reduces to a weighted mean (3.2) in the special case of an additive measure μ. Moreover, consider any linear function $\boldsymbol{x} \mapsto g(\boldsymbol{x}) = \omega_0 + \boldsymbol{\omega}^\top \boldsymbol{x}$ with $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_m)^\top$. This function can also be written in the form

$$\begin{aligned} g(\boldsymbol{x}) &= \omega_0 + \sum_{i=1}^{m} (\omega_i p_i + |\omega_i|(M_i - m_i) z_i) \\ &= \omega_0 + \sum_{i=1}^{m} \omega_i p_i + \sum_{i=1}^{m} |\omega_i|(M_i - m_i) z_i \\ &= \omega_0' + \left(\sum_{i=1}^{m} u_i \right)^{-1} \sum_{i=1}^{m} u_i' z_i \\ &= \gamma \left(\sum_{i=1}^{m} u_i' z_i - \beta \right) \quad , \end{aligned}$$

where $p_i = m_i$ if $\omega_i \geq 0$ and $p_i = M_i$ if $w_i < 0$, $u_i = |\omega_i|(M_i - m_i)$, $\gamma = (\sum_{i=1}^m u_i)^{-1}$, $u_i' = u_i/\gamma$, $\omega_0' = \omega_0 + \sum_{i=1}^m w_i p_i$, $\beta = -\omega_0'/\gamma$. By definition, the u_i' are non-negative and sum up to 1, which means that $\sum_{i=1}^m u_i' z_i$ is a weighted mean of the z_i that can be represented by a Choquet integral.

Probabilistic Thresholding

The model (4.5) can be seen as a two-step process: The first step consists of an assessment of the input $\boldsymbol{x}$ in terms of a utility degree

$$u = U(\boldsymbol{x}) = C_\mu(f_{\boldsymbol{x}}) \in [0, 1].$$

Then, in the second step, a discrete choice (yes/no decision) is made on the basis of this utility. Roughly speaking, this is done through a "probabilistic thresholding" at the utility threshold β. If $U(\boldsymbol{x}) > \beta$, then the decision tends to be positive, whereas if $U(\boldsymbol{x}) < \beta$, it tends to be negative. The precision of this decision is determined by the parameter (see 4.1): For large γ, the decision function converges toward the step function $u \to \mathbb{I}(u > \beta)$, jumping from 0 to 1 at β. For small γ, this function is smooth, and there is a certain probability to violate the threshold rule $u \to \mathbb{I}(u > \beta)$. This might be due to the fact that, despite being important for

decision making, some properties of the instances to be classified are not captured by the utility function. In that case, the utility $U(\boldsymbol{x})$, estimated on the basis of the given attributes, is not a perfect predictor for the decision eventually made. Thus, the parameter γ can also be seen as an indicator of the quality of the classification model.

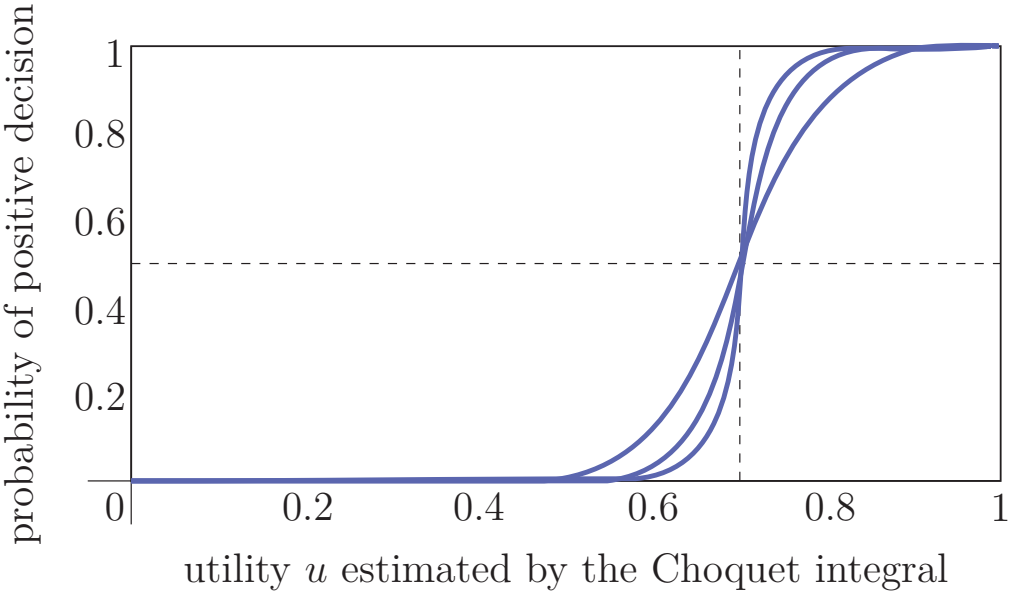

Figure 4.1: Probability of a positive decision, $\mathbf{P}(y = 1 \mid \boldsymbol{x})$, as a function of the estimated degree of utility, $u = U(\boldsymbol{x})$, for a threshold $\beta = 0.7$ and different values of γ.

4.1.3 Maximum Likelihood Estimation

Like the previous case, assume some observations with corresponding responses are given, where the observations are independent identically distributed:

$$\mathcal{D} = \Big\{(\boldsymbol{x}_i, y_i)\Big\}_{i=1}^{n} \subset \mathbb{R}^m \times \Big\{0, 1\Big\} \ .$$

So the goal is to find the parameters with respect to the *choquistic regression* model to derive a proper generalization. As in the case of a linear logistic regression we are interested in optimal parameters. The likelihood function is the same in (2.8), where in this case the probability function is given by (4.5). The log likelihood function for the *choquistic regression* case can be formulated as follows:

$$l(\boldsymbol{m}, \gamma, \beta) = \log \mathbf{P}(\boldsymbol{m}, \gamma, \beta) = \log \left(\prod_{i=1}^{n} \mathbf{P}(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{m}, \beta, \gamma) \right) \tag{4.6}$$

$$= \sum_{i=1}^{n} \left\{ y^{(i)} \log \pi_c^{(i)} + \left(1 - y^{(i)}\right) \log \left(1 - \pi_c^{(i)}\right) \right\} \ . \tag{4.7}$$

In this case again the log-likelihood (4.6) is convex with respect to $\boldsymbol{w}, \gamma$ and β. The basic reason is that, the log likelihood function can be considered the same like in the linear case, but with more attributes. Indeed the inner function namely, $(\mathcal{C}_{\boldsymbol{m}}(\boldsymbol{x}_i) - \beta)$ can be written in a linear form by transferring the parameters. Therefore referring to [15, 96] the log likelihood function has a semidefinite Hessian matrix. Hence the unique solution can be found with a conventional constrained optimization procedure. The parameters can be found by the maximum likelihood principle as follows:

$$\max_{\boldsymbol{m},\gamma,\beta} \left\{ -(1-y)\,\gamma \sum_{i=1}^{n} (\mathcal{C}_{\boldsymbol{m}}(\boldsymbol{x}_i) - \beta) - \sum_{i=1}^{n} \log \left[1 + \exp\left(-\gamma\, \mathcal{C}_{\boldsymbol{m}}(\boldsymbol{x}_i) - \beta\right) \right] \right\} \tag{4.8}$$

s.t.

$$0 \leq \beta \leq 1 \tag{4.9}$$

$$0 < \gamma \tag{4.10}$$

$$\sum_{T \subseteq C} \boldsymbol{m}(T) = 1 \tag{4.11}$$

$$\sum_{B \subseteq A \setminus \{c_i\}} \boldsymbol{m}(B \cup \{c_i\}) \geq 0 \quad \forall A \subseteq C, c_i \in C \tag{4.12}$$

However, the function in (4.8) is convex but additionally there are several linear constraints to assure monotonicity. Theses constraints make the optimization problem certainly more difficult. To solve this convex constrained optimization problem, the Lagrangian method or interior-point methods [81] can be employed.

4.2 Algorithms for Learning Monotone Ordinal Classifiers

Now suppose that aside from two opportunities possible, namely accept and reject for the assessing of articles, there is also another opportunity, namely *reject with encouragement to resubmit*. As can be seen clearly, such cases cannot be modeled by a simple binary logistic regression. In order to extend the idea of binary logistic regression, ordinal logistic regression comes into play.

In many applications in machine learning inheritably, there is a natural order, i.e., there is a total order between classes. So this kind of problem can be taken into account as a special kind of multinomial case and called *ordinal class classification*. From a machine learning point of view, taking into consideration such information, so called *prior knowledge*, sometimes improves the quality of generalization and hence enhances the accuracy of prediction. So the crucial question is how one can apply such information to improve the precision of the model? In the following discussion, one common approach to tackle the ordinal classification problem, called ordinal logistic regression, will be introduced.

4.2.1 Ordinal Logistic Regression

In the binary case, as mentioned in Subsection 4.1.1, the logistic regression models the probability of the positive class (and hence of the negative class) as a linear (affine) function of the input attributes. More specifically, since a linear function does not necessarily produce values in the unit interval, the response is defined as a generalized linear model, namely in terms of the logarithm of the probability ratio:

$$\log\left(\frac{\mathbf{P}(y=1\,|\,\boldsymbol{x})}{\mathbf{P}(y=0\,|\,\boldsymbol{x})}\right) = \omega_0 + \boldsymbol{\omega}^\top \boldsymbol{x} \ , \tag{4.13}$$

where $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_m)^\top \in \mathbb{R}^m$ is a vector of regression coefficients and $\omega_0 \in \mathbb{R}$ a constant bias (the intercept).

Now, consider the case of an ordinal classification, where K ordered classes are given, namely $y_1 \prec \ldots \prec y_K$. The idea of ordinal logistic regression is to reduce the corresponding classification problem to the binary case while taking into account (and actually exploiting) the class order. To this end, it models a probability ratio similar to (4.13), but this time for the *cumulative distribution*:

$$\log\left(\frac{\pi_k(\boldsymbol{x})}{1-\pi_k(\boldsymbol{x})}\right) = \beta_k + \boldsymbol{\omega}^\top \boldsymbol{x} \tag{4.14}$$

for $k \in [K-1] = \{1, \ldots, K-1\}$, where

$$\pi_k(\boldsymbol{x}) = \mathbf{P}(y > y_k \mid \boldsymbol{x}) \tag{4.15}$$

is the (conditional) probability that the class y observed for $\boldsymbol{x}$ is at least y_k; correspondingly,

$$1 - \pi_k(\boldsymbol{x}) = \mathbf{P}(y \leq y_k \mid \boldsymbol{x}) \tag{4.16}$$

is the probability that the class y is less than y_k. Obviously, the left-hand side in (4.14) is nondecreasing in k. Therefore, since the right-hand side only differs in the intercepts (thresholds) β_k, we need to impose the condition

$$\beta_1 \leq \beta_2 \leq \ldots \leq \beta_{K-1} \ .$$

From (4.14), one derives

$$\pi_k(\boldsymbol{x}) = \mathbf{P}(y > y_k \mid \boldsymbol{x}) = \frac{\exp(\beta_k)\exp(\boldsymbol{\omega}^T\boldsymbol{x})}{1+\exp(\beta_k)\exp(\boldsymbol{\omega}^T\boldsymbol{x})}$$

Moreover, exploiting the definition of the cumulative distribution, the class probabilities can be derived as

$$\begin{aligned} &\mathbf{P}(y = y_k \mid \boldsymbol{x}) \\ &= \mathbf{P}(y > y_{k-1}|\boldsymbol{x}) - \mathbf{P}(y > y_k|\boldsymbol{x}) \\ &= \pi_{k-1}(\boldsymbol{x}) - \pi_k(\boldsymbol{x}) \end{aligned}$$

for $k \in [K]$ (where $\pi_k(\boldsymbol{x}) = 1$ and $\pi_0(\boldsymbol{x}) = 0$ by definition).

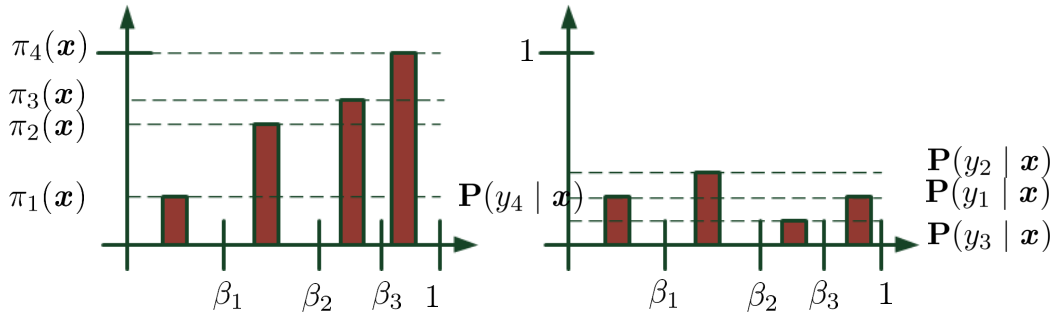

Figure 4.2: Illustration of the ordinal logistic regression model for $\mathcal{Y} = \{y_1, y_2, y_3, y_4\}$: Class assignment via hard thresholding (left) versus probabilistic classification. The cumulative distribution $y_k \mapsto \pi_k(\boldsymbol{x})$ is shown in the middle, the probability distribution $y_k \mapsto \mathbf{P}(y = y_k \mid \boldsymbol{x}) = \pi_k(\boldsymbol{x}) - \pi_{k-1}(\boldsymbol{x})$ on the right.

4.2.2 Maximum Likelihood Estimation

The model (4.14) has several degrees of freedom: The weights parameters ($\boldsymbol{\omega}$) and intercepts parameters ($\boldsymbol{\beta}$). The goal of learning is to identify these degrees of freedom on the basis of the training data. Assume some observations with corresponding responses are given, where the observations are independent identically distributed as follows:

$$\mathcal{D} = \Big\{(\boldsymbol{x}_i, y_i)\Big\}_{i=1}^{n} \subset \mathbb{R}^m \times \Big\{y_1, \ldots, y_K\Big\} \ .$$

Like in the case of a standard logistic regression, it is possible to harness the maximum likelihood (ML) principle for this purpose. Given a set of training data, the estimation of the parameters $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{K-1})$ and $\boldsymbol{\omega}$ is then accomplished through maximum likelihood estimation, i.e., by maximizing the log-likelihood

$$l(\beta, \boldsymbol{\omega}) = \sum_{i=1}^{n} \log \mathbf{P}(y_i \mid \boldsymbol{x}_i) \tag{4.17}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{I}(y_i = y_k) \log \Big(\pi_k(\boldsymbol{x}_i) - \pi_{k-1}(\boldsymbol{x}_i)\Big) \ , \tag{4.18}$$

where $\mathbb{I} : \mathcal{Y} \to \{0, 1\}$ is the indicator function. Furthermore $\beta_1 \leq \ldots \leq \beta_{K-1}$. More precisely, the constrained optimization problem can be formalized as follows:

$$\max_{\boldsymbol{\omega},\boldsymbol{\beta}} \sum_{i=1}^{n}\sum_{k=1}^{K} \mathbb{I}(y_i = y_k) \log\Big(\pi_k(\boldsymbol{x}_i) - \pi_{k-1}(\boldsymbol{x}_i)\Big) \tag{4.19}$$

s.t.

$$0 \leq \beta_1 \leq \beta_2 \leq \ldots \leq \beta_{K-1} \ .$$

Note that the function in (4.19) is concave with respect to $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_m)$ and to $\boldsymbol{\beta} = (\beta_1 \ldots, \beta_{K-1})$. In the following, we show that the Hessian matrix is negative semi-definite with respect to the $(\boldsymbol{\omega}, \boldsymbol{\beta})$. Actually the basic idea is to show that all second partial derivatives are negative with respect to the $(\boldsymbol{\omega}, \boldsymbol{\beta})$. This follows to conclude that Hessian matrix is negative semi-definite with respect to the $(\boldsymbol{\omega}, \boldsymbol{\beta})$. To this end assume

$$\pi_k(\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{\omega}^\top \boldsymbol{x} - \beta_k)} \ ,$$

and also

$$\begin{aligned} F(\boldsymbol{\omega}, \boldsymbol{\beta}) &= \pi_k(\boldsymbol{x}) - \pi_{k-1}(\boldsymbol{x}) \\ &= \frac{1}{1 + \exp(-\boldsymbol{\omega}^\top \boldsymbol{x} - \beta_k)} - \frac{1}{1 + \exp(-\boldsymbol{\omega}^\top \boldsymbol{x} - \beta_{k-1})} \ . \end{aligned}$$

Since the sum of convex functions are convex, it is enough to show that the $-\log\big(F(\boldsymbol{\omega}, \boldsymbol{\beta})\big)$ is convex. Let simplify the $\log\big(F(\boldsymbol{\omega}, \boldsymbol{\beta})\big)$:

$$\begin{aligned} G(\boldsymbol{\omega}, \boldsymbol{\beta}) = \log\big(F(\boldsymbol{\omega}, \boldsymbol{\beta})\big) = -\boldsymbol{\omega}^\top \boldsymbol{x} + \Big\{ \exp(-\beta_{k-1}) - \exp(-\beta_k) \Big\} \\ - \log\Big\{ 1 + \exp(-f(\boldsymbol{\omega}) - \beta_{k-1}) + \exp(-f(\boldsymbol{\omega}) - \beta_k) \\ + \exp(-2f(\boldsymbol{\omega}) - \beta_{k-1} - \beta_k) \Big\} \ . \end{aligned}$$

Note that the function $G(\boldsymbol{\omega}, \boldsymbol{\beta})$ is well defined if and only if $\beta_k > \beta_{k-1}$. Without a loss of generality, one can consider $\beta_k = \beta_{k-1} + \theta^2$. In the first step we show that $G(\boldsymbol{\omega}, \boldsymbol{\beta})$ is concave with respect to $(\omega_1, \ldots, \omega_m)$. Note that in addition, we assume $\boldsymbol{x} \in \mathbb{R}^m_+$. In the following calculations, the first partial derivative with respect to ω_i is computed as follows:

$$\begin{aligned} &\frac{\partial G(\boldsymbol{\omega}, \boldsymbol{\beta})}{\partial w_i} = -x_i \\ &+ \frac{x_i \exp(-f(\boldsymbol{\omega}) - \beta_{k-1}) + x_i \exp(-f(\boldsymbol{\omega}) - \beta_k) + 2x_i \exp(-2f(\boldsymbol{\omega}) - \beta_{k-1} - \beta_k)}{1 + \exp(-f(\boldsymbol{\omega}) - \beta_{k-1}) + \exp(-f(\boldsymbol{\omega}) - \beta_k) + \exp(-2f(\boldsymbol{\omega}) - \beta_{k-1} - \beta_k)} \ , \end{aligned}$$

where in addition $f(\boldsymbol{\omega}) = \boldsymbol{\omega}^\top \boldsymbol{x}$. Assuming

$$\begin{aligned}
D(\boldsymbol{\omega}, \boldsymbol{\beta}) :=& 1 + \exp(-f(\boldsymbol{\omega}) - \beta_{k-1}) + \exp(-f(\boldsymbol{\omega}) - \beta_k) \\
&+ \exp(-2f(\boldsymbol{\omega}) - \beta_{k-1} - \beta_k) \\
N(\boldsymbol{\omega}, \boldsymbol{\beta}) :=& x_i \exp(-f(\boldsymbol{\omega}) + \beta_{k-1}) - x_i \exp(-f(\boldsymbol{\omega}) - \beta_k) \\
&+ 2x_i \exp(-2f(\boldsymbol{\omega}) - \beta_{k-1} - \beta_k) \\
Exp_k(\boldsymbol{\omega}, \boldsymbol{\beta}) :=& \exp(-f(\boldsymbol{\omega}) - \beta_k) \\
Exp_{k,k-1}(\boldsymbol{\omega}, \boldsymbol{\beta}) :=& \exp(-2f(\boldsymbol{\omega}) - \beta_{k-1} - \beta_k) \ ,
\end{aligned}$$

the second partial derivative with respect to ω_j is equal to:

$$\begin{aligned}
&\frac{\partial^2 G(\boldsymbol{\omega}, \boldsymbol{\beta})}{\partial w_i \partial w_j} = \\
&- \frac{\left\{ x_i x_j Exp_{k-1}(\boldsymbol{\omega}, \boldsymbol{\beta}) + x_i x_j Exp_k(\boldsymbol{\omega}, \boldsymbol{\beta}) + 2x_i x_j Exp_{k,k-1}(\boldsymbol{\omega}, \boldsymbol{\beta}) \right\} \cdot D(\boldsymbol{\omega}, \boldsymbol{\beta})}{D(\boldsymbol{\omega}, \boldsymbol{\beta})^2} \\
&- \frac{\left\{ x_j Exp_{k-1}(\boldsymbol{\omega}, \boldsymbol{\beta}) + x_j Exp_k(\boldsymbol{\omega}, \boldsymbol{\beta}) + 2x_j Exp_{k,k-1}(\boldsymbol{\omega}, \boldsymbol{\beta}) \right\} \cdot N(\boldsymbol{\omega}, \boldsymbol{\beta})}{D(\boldsymbol{\omega}, \boldsymbol{\beta})^2} \ .
\end{aligned}$$

Hence

$$\forall\, i, j \qquad \frac{\partial^2 G(\boldsymbol{\omega}, \boldsymbol{\beta})}{\partial w_i \partial w_j} \leq 0 \ .$$

Note that $\boldsymbol{x} \in \mathbb{R}_+^m$.
For the variables $\beta_k, \beta_{k-1} \ \ (2 \leq k \leq K-1)$ the first and second derivatives are as follows:

$$\begin{aligned}
&\frac{\partial G(\boldsymbol{\omega}, \boldsymbol{\beta})}{\partial \beta_k} = \exp(-\beta_k) \\
&+ \frac{\exp(-f(\boldsymbol{\omega}) - \beta_k) + \exp(-2f(\boldsymbol{\omega}) - \beta_{k-1} - \beta_k)}{1 + \exp(-f(\boldsymbol{\omega}) - \beta_{k-1}) + \exp(-f(\boldsymbol{\omega}) - \beta_k) + \exp(-2f(\boldsymbol{\omega}) - \beta_{k-1} - \beta_k)} \ .
\end{aligned}$$

$$\frac{\partial^2 G(\boldsymbol{\omega},\boldsymbol{\beta})}{\partial \beta_k^2} = -\exp(-\beta_k) - \frac{\left\{\exp(f(\boldsymbol{\omega})-\beta_k)+\exp(-2f(\boldsymbol{\omega})-\beta_{k-1}-\beta_k)\right\}\cdot D(\boldsymbol{\omega},\boldsymbol{\beta})}{D(\boldsymbol{\omega},\boldsymbol{\beta})^2} - \frac{\left\{-\exp(f(\boldsymbol{\omega})-\beta_k)-\exp(-2f(\boldsymbol{\omega})-\beta_{k-1}-\beta_k)\right\}^2}{D(\boldsymbol{\omega},\boldsymbol{\beta})^2}$$

$$\frac{\partial^2 G(\boldsymbol{\omega},\boldsymbol{\beta})}{\partial \beta_k \partial \beta_{k-1}} = -\frac{\exp(-2f(\boldsymbol{\omega})-\beta_{k-1}-\beta_k)\cdot D(\boldsymbol{\omega},\boldsymbol{\beta})}{D(\boldsymbol{\omega},\boldsymbol{\beta})^2} - \left\{\frac{\left\{-\exp(f(\boldsymbol{\omega})-\beta_{k-1})-\exp(-2f(\boldsymbol{\omega})-\beta_{k-1}-\beta_k)\right\}}{D(\boldsymbol{\omega},\boldsymbol{\beta})} \times \frac{\left\{-\exp(f(\boldsymbol{\omega})-\beta_k)-\exp(-2f(\boldsymbol{\omega})-\beta_{k-1}-\beta_k)\right\}}{D(\boldsymbol{\omega},\boldsymbol{\beta})}\right\} .$$

Hence

$$\forall\ k \quad 1 \le k \le K-1, \ \ \frac{\partial^2 G(\boldsymbol{\omega},\boldsymbol{\beta})}{\partial \beta_k^2} \le 0$$

$$\forall\ k \quad 2 \le k \le K-1, \ \ \frac{\partial^2 G(\boldsymbol{\omega},\boldsymbol{\beta})}{\partial \beta_k \partial \beta_{k-1}} \le 0 \ .$$

Also

$$\frac{\partial^2 G(\boldsymbol{\omega},\boldsymbol{\beta})}{\partial \omega_i \partial \beta_k} = - \frac{\left\{x_i \exp(-f(\boldsymbol{\omega})-\beta_k)+2x_i\exp(-2f(\boldsymbol{\omega})-\beta_{k-1}-\beta_k)\right\}\cdot D(\boldsymbol{\omega},\boldsymbol{\beta})}{D(\boldsymbol{\omega},\boldsymbol{\beta})^2} - \frac{\left\{\exp(-f(\boldsymbol{\omega})-\beta_k)+\exp(-2f(\boldsymbol{\omega})-\beta_{k-1}-\beta_k\right\}\cdot N(\boldsymbol{\omega},\boldsymbol{\beta})}{D(\boldsymbol{\omega},\boldsymbol{\beta})^2}$$

It is easy to show that,

$$\forall k \ \ k \in \{1,\ldots,K-1\}, \ \forall\ i \ \ i \in \{1,\ldots,m\}, \ \ \frac{\partial^2 G(\boldsymbol{\omega},\boldsymbol{\beta})}{\partial \beta_k \partial \omega_i} \le 0 \ .$$

Since all second derivatives with respect to $(\boldsymbol{\omega},\boldsymbol{\beta})$ are negative then, the Hessian matrix is negative semi definite, and therefore the logarithm of the likelihood function is concave with respect to $(\boldsymbol{\omega},\boldsymbol{\beta})$. On the contrary, the function $-G(\boldsymbol{\omega},\boldsymbol{\beta})$ is minimized, therefore is convex respect to $(\boldsymbol{\omega},\boldsymbol{\beta})$.

4.2.3 Ordinal Choquistic Regression

In order to model non-linear dependencies between predictor variables and response and to take interactions between predictors into account, it is proposed to extend the ordinal logistic regression model by replacing the (affine) linear function $\boldsymbol{x} \mapsto \beta_k + \boldsymbol{w}^\intercal \boldsymbol{x}$ in (4.14) by the Choquet integral. More specifically, we propose the following model

$$\log\left(\frac{\pi_k(\boldsymbol{x})}{1-\pi_k(\boldsymbol{x})}\right) = \gamma\Big(\mathcal{C}_\mu(f_{\boldsymbol{x}}) - \beta_k\Big),$$

where $\mathcal{C}_\mu(f_{\boldsymbol{x}})$ is the Choquet integral (with respect to the measure μ) of the evaluation function

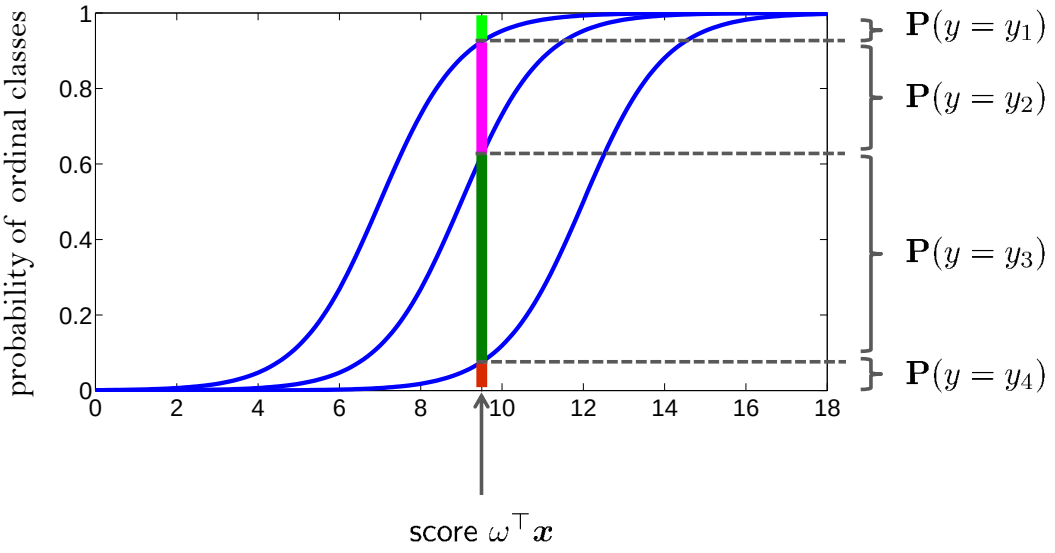

Figure 4.3: The Illustration of the ordinal choquistic regression for 4 ordinal classes, with same γ value and different β values

$$f_{\boldsymbol{x}} : \{c_1, \ldots, c_m\} \to [0, 1]$$

that maps each attribute c_i to a value $x_i = f_{\boldsymbol{x}}(c_i)$; $\gamma \geq 0$ and $\beta_1, \ldots, \beta_{K-1}$ are real constraints such that $0 = \beta_1 \leq \beta_2 \leq \ldots \leq \beta_{K-1} \leq 1$. In Figure 4.3, the probability of ordinal classes for a given instance $\boldsymbol{x}$ is shown.

4.2.4 Maximum Likelihood Estimation

Assume some observations with corresponding responses are given, in which the observations are independent identically distributed:

$$\mathcal{D} = \left\{(\boldsymbol{x}^{(i)}, y^{(i)})\right\}_{i=1}^{n} \subset \mathbb{R}^m \times \left\{y_1, \ldots, y_K\right\} \ .$$

So the goal is to find the proper generalization.

The model (4.17) has several degrees of freedom: The fuzzy measure μ (Möbius transform $\boldsymbol{m} = \boldsymbol{m}_\mu$) determines the (latent) utility function, while the utility thresholds $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{K-1})$ and the scaling parameter γ determine the discrete choice model. The goal of learning is to identify these degrees of freedom on the basis of the training data. Like in the case of standard logistic regression, it is possible to harness the maximum likelihood (ML) principle for this purpose. The log-likelihood of the parameters can be written as

$$\begin{aligned} l(\boldsymbol{m}, \gamma, \beta) = \log \mathbf{P}(\mathcal{D} \mid \boldsymbol{m}, \gamma, \boldsymbol{\beta}) &= \log \left(\prod_{i=1}^{n} \mathbf{P}(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{m}, \boldsymbol{\beta}, \gamma) \right) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{I}\left(y^{(i)} = y_k\right) \log \pi_k^*\Big(\boldsymbol{x}^{(i)}, \boldsymbol{m}, \boldsymbol{\beta}, \gamma\Big) \ , \end{aligned}$$

where

$$\begin{aligned} \pi_k^*(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{\beta}, \gamma) = &\frac{\exp(-\gamma\beta_k)\exp(\gamma \mathcal{C}_\mu(f_{\boldsymbol{x}}))}{1 + \exp(-\gamma\beta_k)\exp(\gamma \mathcal{C}_\mu(f_{\boldsymbol{x}}))} \\ &- \frac{\exp(-\gamma\beta_{k-1})\exp(\gamma \mathcal{C}_\mu(f_{\boldsymbol{x}}))}{1 + \exp(-\gamma\beta_{k-1})\exp(\gamma \mathcal{C}_\mu(f_{\boldsymbol{x}}))} \ . \end{aligned}$$

The log likelihood function also in this case can be taken into account as a log likelihood function in an ordinal logistic regression case. Seen from this view, the log likelihood is indeed a concave function, however, in the presence of several constraints as well. Therefore, it is counted as a constrained optimization problem. In principle, maximization of the log-likelihood can hence be accomplished by means of standard gradient-based optimization methods. However, since we have to assure that μ is a proper fuzzy measure and, hence, that m guarantees the corresponding

monotonicity and boundary conditions, we actually need to solve a constrained optimization problem. Namely, the optimization under the following conditions (recall that $C = \{c_1, \ldots, c_m\}$ denotes the set of predictor variables):

$$\max_{\boldsymbol{m},\gamma,\beta} \left\{ \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{I}\left(y^{(i)} = y_k\right) \log \pi_k^*\Big(\boldsymbol{x}^{(i)}, \boldsymbol{m}, \boldsymbol{\beta}, \gamma\Big) \right\} \tag{4.20}$$

s.t.

$$0 \leq \beta_1 \leq \beta_2 \leq \ldots \leq \beta_{K-1} \leq 1 \tag{4.21}$$

$$0 < \gamma \tag{4.22}$$

$$\sum_{T \subseteq C} \boldsymbol{m}(T) = 1 \tag{4.23}$$

$$\sum_{B \subseteq A \setminus \{c_i\}} \boldsymbol{m}(B \cup \{c_i\}) \geq 0 \quad \forall A \subseteq C, c_i \in C \tag{4.24}$$

Note that, the log likelihood function with respect to $(\boldsymbol{\omega}, \boldsymbol{\beta})$ is concave. The reason is the same as in the previous section. One can easily imagine, the ordinal choquistic regression without constraints has exactly the same structure as an ordinal logistic regression. However, in addition there are several constraints (linear) to assure monotonicity. Theses constraints make the optimization problem certainly more difficult. To this end, we used the `fmincon` function implemented in the optimization toolbox of MATLAB. This method is based on a sequential quadratic programming (SQP) approach.

4.3 Related Researches

However, the binary classifiers and ordinal classifiers in essence are used for different goals, but since this chapter solely focuses on monotone classifiers, the ordinal classifiers can be seen as an extension of binary classifiers. Seen from this point of view, the approaches related to ordinal classifiers can also tackle the binary classification problem. In order to investigate in more detail monotone classifiers, this section is allocated to present existing methods and approaches, mainly with respect to the Choquet integral. In this regard, this section begins by describing the approaches for the binary case.

Before continuing the topic, note that from a model class point of view, one can distinguish between deterministic and probabilistic predictions. In deterministic

predictions the range of classifiers is $\{0, 1\}$, namely classifier CL_d is assumed as follows:

$$CL_d : \mathcal{X} \to \{0, 1\}$$

however at probabilistic case, the range of classifier CL_p is $[0, 1]$, namely

$$CL_p : \mathcal{X} \to [0, 1] \ ,$$

where $\mathcal{X}$ is domain of attributes. In this regard, we take into consideration rule models [38], which consider conjunctive statements as rules; k-nearest neighbor [25, 40], which makes prediction based on k closest seen samples; decision trees [38, 89], which go down (up) a tree successively and make the prediction at the end of a leaf; support vector machines [108], in which the so-called hyper plane maximizes the distance between two different classes, as deterministic approaches, whereas the logistic regression model [16, 58], Bayesian regression [38], models the dependency between input/output in a probabilistic way; kernel logistic regression [114], which uses the logistic regression idea in a kernel framework as probabilistic approaches. In the following discussion, we start with the approaches based on the Choquet integral for binary classification in a deterministic framework.

Although the Choquet integral has been widely applied as an aggregation operator in MCDM [42, 47, 52, 104], it has been used much less in the field of machine learning so far. There are, however, a few notable exceptions. First, the problem of extracting a Choquet integral (or, more precisely, the non-additive measure on which it is defined) in a data-driven way has been addressed in the literature. Essentially, this is a parameter identification problem, which is commonly formalized as a constraint optimization problem, for example using the sum of squared errors as an objective function [48, 105]. To this end, [77] proposed an approach based on the use of quadratic forms, while an alternative heuristic, gradient-based method called HLMS (Heuristic Least Mean Squares) was introduced in [43]. Besides, genetic algorithms have been used as a tool for parameter optimization [63, 64]. Some mathematical results regarding this optimization problem can be found in [61, 62]. In particular methods for binary classification based on the Choquet integral were developed in [53] and [111]. In [53], Grabisch et al. essentially employed the Choquet integral as a fusion operator in this context. For an instance $\boldsymbol{x} = (x_1, \ldots, x_m)$, let $\phi_i^{(j)}(\boldsymbol{x})$ express a measure of confidence (provided by feature c_i) that $\boldsymbol{x}$ belongs to class $j \in \{0, 1\}$. Grabisch defines the global confidence for class j as an aggregation of these confidence degrees:

$$\phi_{\mu^{(j)}}(\boldsymbol{x}) \stackrel{\text{df}}{=} \mathcal{C}_{\mu^{(j)}} \left(\phi_1^{(j)}(\boldsymbol{x}), \ldots, \phi_m^{(j)}(\boldsymbol{x}) \right) \ ,$$

where $\mathcal{C}_\mu$ denotes the discrete Choquet integral with respect to the fuzzy measure μ. Eventually, the class with the highest global confidence is predicted as an output. Here, the fuzzy measures $\mu^{(0)}$ and $\mu^{(1)}$ express the importance of the features and groups of features in the classification process. The $\phi_i^{(j)}$ are assumed to be derived by means of a conventional parametric or nonparametric probability density estimation method, subsequent to suitable normalization. The identification of the fusion operator is then reduced to the identification (or learning) of the fuzzy measures $\mu^{(0)}$ and $\mu^{(1)}$ with $2(2^m - 2)$ coefficients. To this end, Grabisch minimizes the empirical squared error loss

$$\begin{aligned} J = &\sum_{\boldsymbol{x} \in T_0} \left(\phi_{\mu^{(0)}}(\boldsymbol{x}) - \phi_{\mu^{(1)}}(\boldsymbol{x}) - 1\right)^2 \\ &+ \sum_{\boldsymbol{x} \in T_1} \left(\phi_{\mu^{(1)}}(\boldsymbol{x}) - \phi_{\mu^{(0)}}(\boldsymbol{x}) - 1\right)^2 \quad , \end{aligned} \tag{4.25}$$

i.e., the sum of squared differences between predicted and given output values, using standard optimization routines (T_0 and T_1 denote, respectively, the set of observed negative and positive examples). However, taking the fact into account that, this model provides finally two different fuzzy measures ($\mu^{(0)}, \mu^{(1)}$) causing difficulties from an interpretational point of view. Assume the Shapley index and interaction index for each measure are already computed. Since they are considered with respect to the specific classes, namely positive and negative classes individually, they are not representative for whole observations. However, from a choquistic regression point of view, the model provides a unique fuzzy measure given observations. Therefore the interpretation is also valid for whole data.

Yan, Wang and Chen [111] tackle a quite similar problem, albeit using another optimization criterion (which can be seen as a kind of relaxed class separability criterion). Besides, the authors define the Choquet integral based on a so-called signed non-additive measure [79]. A signed non-additive measure μ defined on $C = \{c_1, \ldots, c_m\}$ is a set function $\mu : P(C) \to (-\infty, \infty)$ satisfying $\mu(\emptyset) = 0$. In other words, signed non-additive measure does not satisfy monotonicity constraints. Based on signed non-additive measure, the Choquet integral can be computed as follows:

$$(c) \int f d\mu = \sum_{j=1}^{2^m - 1} z_j . \mu_j \quad ,$$

where

$$z_j = \begin{cases} \min f(x_i)_{i:a_{ij}} - \max f(x_i)_{i:b_{ij}} & \text{if} > 0 \ \text{ or } \ j = 2^m - 1 \\ 0 & \text{otherwise} \end{cases}$$

with $a_{ij} = \text{frc}(j/2^i) \in [0.5, 1)$ and $b_{ij} = \text{frc}(j/2^i) \in [0, 0.5)$. Here, $\text{frc}(j/2^i)$ is the fractional part of $j/2^i$ and the maximum operation on the empty set is zero. For binary classification purposes they assumed the following setting:

$$\max_{\boldsymbol{\beta},\mu} \sum_{i=1}^{n} \beta_i$$

such that

$$(c) \int f d\mu - \boldsymbol{b} \leq \boldsymbol{\beta} \qquad \text{if the case belongs to the first class}$$

$$(c) \int f d\mu - \boldsymbol{b} \geq -\boldsymbol{\beta} \qquad \text{if the case belongs to the second class}$$

$$\beta_i \geq 0$$

where $\boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix}$ and there are in total n records. The obvious advantage is that here there are not any monotonicity constraints.

So far it was assumed that the classes are binary. In the following discussion, the approaches with respect to ordinal class classification are presented. Ordinal classification, a special type of multi-class classification in which the class labels are linearly ordered (e.g., a paper submitted for publication can be labeled as *accept*, *weak accept*, *weak reject*, or *reject*). The main idea of the ordinal class classification problem has been introduced in Section (4.2). Note that in this case K classes are given as follows:

$$y_1 \prec \ldots \prec y_K \ .$$

Frank and Hall in [39] proposed a meta approach based on a decomposition idea. Basically in order to tackle the ordinal class classification problem, they decompose

the K - ordinal classes into $K-1$ independent binary class problems and solve them fully separately. Needless to say, after decomposition all (ordinal) binary classifiers can tackle the problem, which can be seen as an advantage. However the approach ignores some of the dependencies between ordinal classes, due to the decomposition step, i.e., there is a danger to loses some of information and interdependencies. The ordinal logistic regression (OLR) is the most common approach, which solves the problem in a probabilistic framework. This approach was described in detail in Section 4.2.1. The two mentioned approaches are common approaches to tackling the ordinal class classification problem. In the following discussion, we will focus more on the approaches based on the Choquet integral. Historically the Choquet integral was mostly used as an aggregation operator in decision making, and in particular as a tool for preference elicitation [2, 4, 42, 104], while it is less applied for the purpose of learning. However, while for ordinal classification purposes, the Choquet integral has not been used a great deal, there are a few notable exceptions: For ordinal class classification purposes, Grabisch et al. [49, 52, 54] consider input data of the following kind: A reference set of objects $A = \{\mathcal{O}_1, \ldots, \mathcal{O}_l\}$ and a set of criteria $C = \{c_1, \ldots, c_m\}$; a table of individual scores (performances) z_{ki} ($\mathcal{O}_k \in A$, $c_i \in C$); a partial preorder $\succeq_A$ on A (partial ranking of the objects on a global basis); a partial preorder $\succeq_C$ on C (partial ranking of the criteria); a partial preorder $\succeq_P$ on the set of pairs of criteria (partial ranking of interaction); the sign of interaction between selected pairs of criteria, reflecting synergy, independence or redundancy. All this information can be translated into linear equalities or inequalities between the weights of the underlying fuzzy measure μ. This measure is then identified based on a constraint optimization problem, using as an objective function a criterion that resembles very much the so-called margin principle in machine learning. The method itself, however, is more oriented towards decision making and less suitable for machine learning applications. In particular, it is not tolerant toward noise in the data, and, in terms of complexity, does not scale well with the size of the data. In addition, specifically for (ordinal) binary classification purposes, the whole constraints with respect to different classes must always be considered, which makes the optimization problem more sophisticated. In (ordinal) choquistic/logistic regression, the information about classes is a part of the objective function (4.20), and there is no need to add auxiliary constraints. This is indeed a profit to reduce the complexity of the learning problem.

Torra also in [104] used the above setting to tackle the ordinal class classification problem. To this end, he supposed that the observations are ranked. Namely instead of considering ordinal classes he assumes the ordering for observations. Then the approach tries to estimate the suitable parameters by minimizing the square error

with additional constraints to ensuring monotonicity and can be solved by quadratic programming. This ordering can be substituted by ordinal classes, hence the approach can tackle the ordinal class classification problem too. Let us have a look at OLR more precisely. The OLR has two types of parameters; weight parameters $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_m)$ and intercept parameters $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{K-1})$. The weight parameters and intercept parameters are estimated in the OLR case during the learning process, whereas in the model proposed by Torra the intercept parameters are given in advance. It is clear that if in advance they are determined wrongly, the corresponding results are not truthworthy. Especially for each dataset the intercept parameters must be determined in advance.
In [2, 4] the Choquet integral has been used as a tool for preference elicitation. To this end, particularly in [4], the following setting are assumed:

- pairwise preference on the alternatives,
- the comparison in terms of intensity between pair of alternatives,
- the joint comparison between importancy of criteria and their differences with each other,
- enforcing negative and positive interaction expressing redundancy or synergy,
- the joint comparison of interaction intensity among couples of criteria and their differences between each other.

Then they used linear programming to find the corresponding fuzzy measure, with respect to the above constraints. In [3], the proposed solution to the problem of learning an optimal classification function is framed through margin-maximization.

In [8], Beliakov and James develop a method for classifying journals in the field of pure mathematics, which are rated on an ordinal scale with categories A^+, A, B and C ($C \prec B \prec A \prec A^+$). The classification is done on the basis of five criteria serving as input attributes, namely the number of citations per year, the impact factor, the immediacy index, the total number of articles published, and the cited half-life index (we shall use the same data set in our experiments later on). As a loss function, the authors use the absolute difference between the predicted class and the target (i.e., the loss is $|i - j|$ if the i-th class is predicted although the j-th class would be correct). Basically the authors in this paper used the Choquet integral as a tool for modeling information that may be correlated. The authors applied FMTools, which used the least absolute deviation criterion in order to find the weights of a fuzzy measure. In this case, the obvious disadvantage is the effect

of the class, which has the majority of instances. In an extreme case, the optimal solution has a high tendency for the class that has the majority of instances, which indeed is a wrong bias.

5

Kernel-Based Learning and Support Vector Machines

This chapter is mainly devoted to the learning the Choquet integral in a support vector machines framework. The main advantage of a support vector machine is to solve the problem by linear programming (with additional constraints). Hence, simpler solvers also can solve the problem. Besides, the parameters for the Choquet integral are learnt in a kernel-based framework. To this end, the Choquet kernels in Section 5.2 will be introduced. The Choquet kernel is less complex. In this regard, we discuss exhaustively on the usefulness of the Choquet kernel for classification purposes and corresponding to the k-additivity, different kernels are presented. The basic idea is to learn the weights in dual form. In contrast to primal form (2.4.2) there is no way to guarantee the monotonicity in dual form; in general, the learned weights for the Choquet kernel do not satisfy monotonicity constraints. To overcome this inconsistency several algorithms in Chapter 6, Section 6.4 are introduced. Roughly speaking, the core idea is to modify the learned weights, so that finally they can satisfy monotonicity constraints. As will be clear in the following discussion, defining the Choquet integral in a kernel framework has several advantages, particularly in terms of computational complexity-reduction. In Chapter 7, Subsection 7.8.2 the comparison in terms of running time, regarding the complexity reduction is demonstrated.

Parts of this chapter were already published in [102].

5.1 Learning the Choquet Integral by Employing SVM

5.1.1 Primal Form

In this section we focus on the integration of the Choquet integral into support vector machines. Let us assume some labeled instances, labeled by two different classes, which in addition are assumed be *i.i.d.* given as follows:

$$\Big\{(\boldsymbol{x}_i, y_i)\Big\}_{i=1}^{n} \subset \mathbb{R}^m \times \Big\{-1, +1\Big\} \ .$$

Here the main idea refers to learn the Möbius transform parameters by using support vector machines. The basic idea of support vector machines was given in Subsection 2.4.2. Here the idea is referred to representation in (3.3). Basically this representation can be seen as an inner product between Möbius transform vector and the basis functions. The set of basis functions are:

$$\Big\{ \min_{i \in T}\{x_i\} \ \Big| \ T \subseteq C \Big\}$$

This representation allows us to consider a mapping called $\varphi : \mathbb{R}^m \to \mathbb{R}^{2^m-1}$ as follows:

$$\varphi(\boldsymbol{x}) = \varphi(x_1, \ldots, x_m) = \Big(x_1, \ldots, x_m, \min\{x_1, x_2\}, \ldots, \min\{x_{m-1}, x_m\}, \\ \min\{x_1, x_2, x_3\}, ..., \min\{x_1, \ldots, x_m\}\Big)$$

Seen from this view, given the Möbius transform $\boldsymbol{m}_\mathfrak{J}$ the discrete Chqouet integral is equal to:

$$\mathcal{C}_{\boldsymbol{m}_\mathfrak{J}}(\boldsymbol{x}) = \Big\langle \boldsymbol{m}_\mathfrak{J}, \varphi(\boldsymbol{x}) \Big\rangle \ ,$$

where in addition $\boldsymbol{m}_\mathfrak{J}$ denotes the Möbius transform vector with the following orders:

$$\Big(\boldsymbol{m}(\{c_1\}), \ldots, \boldsymbol{m}(\{c_n\}), \boldsymbol{m}(\{c_1, c_2\}), \ldots, \boldsymbol{m}(\{c_{n-1}, c_n\}), \ldots, \boldsymbol{m}(\{c_1, \ldots, c_n\})\Big).$$

Note that the order in $\boldsymbol{m}_\mathfrak{J}$ is exactly the same in $\varphi(\boldsymbol{x})$. Moreover, note that here solely the pure definition of the discrete Choquet integral is assumed, and therefore

instead of evaluation function $f(\cdot)$, we consider the instance $\boldsymbol{x}$. The inner product representation allows us to consider $\varphi(\boldsymbol{x})$ as a feature mapping in SVM. Therefore the main goal is to learn the Möbius transform parameters. Remember that in order to ensure monotonicity for fuzzy measure (μ) or correspondingly the Möbius transform ($\boldsymbol{m}$) the constraints in (4.12) must be considered. Finally the Möbius transform parameters can be estimated as follows:

$$
\begin{aligned}
&\min_{\boldsymbol{\omega},\boldsymbol{\xi},b} \left\{ \frac{1}{2}\boldsymbol{\omega}^\top \boldsymbol{\omega} + \frac{C}{n}\sum_{i=1}^{n} \xi_i \right\} \\
&\text{s.t.} \\
&y_i\Big(\langle \boldsymbol{\omega}, \varphi(\boldsymbol{x}_i)\rangle + b \Big) \geq 1 - \xi_i, \qquad \forall\, i \in \{1,\ldots,n\} \\
&\xi_i \geq 0 \qquad \forall\, i \in \{1,\ldots,n\} \\
&\sum_{B \subseteq A\setminus\{c_i\}} \boldsymbol{m}(B \cup \{c_i\}) \geq 0 \qquad \forall A \subseteq C, \forall c_i \in C
\end{aligned} \tag{5.1}
$$

where the vector $\boldsymbol{\omega}$ is equal to:

$$
\Big(\boldsymbol{m}(\{c_1\}), \ldots, \boldsymbol{m}(\{c_n\}), \boldsymbol{m}(\{c_1, c_2\}), \ldots, \boldsymbol{m}(\{c_{n-1}, c_n\}), \ldots, \boldsymbol{m}(\{c_1, \ldots, c_n\}) \Big)
$$

corresponding to feature mapping $\varphi(\boldsymbol{x})$. As can be seen, besides of constraints for classes, there are additionally $m2^{m-1}$ constraints in order to guarantee monotonicity. Also the feature mapping itself has $2^m - 1$ components. In fact the complexity of the problem is of course exponential and frankly speaking, for $n \geq 20$ computationally infeasible. Note that, in our setting $\sum_{B\subseteq X} \boldsymbol{m}(B) = 1$ is omitted on purpose (such constraints can be assumed at the end of the learning process).

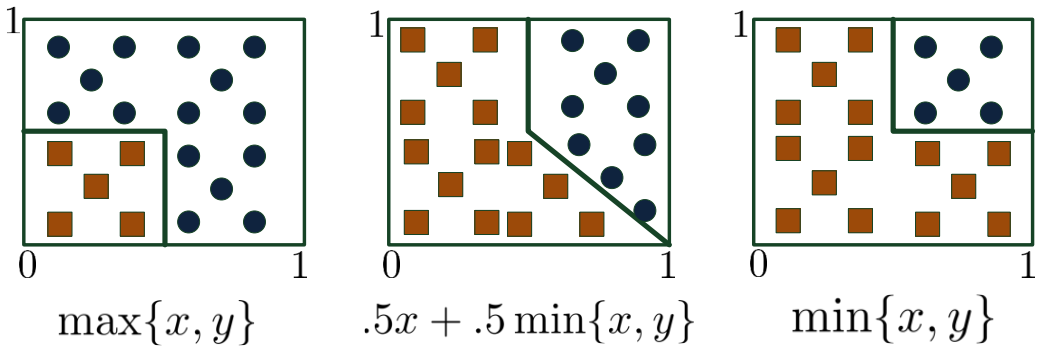

Figure 5.1: Decision boundary for the Choquet integral in the case of $\{(x, y) \in [0, 1]^2 \mid x + y - \min\{x, y\} = \max\{x, y\} > .5\}$, $\{(x, y) \in [0, 1]^2 \mid .5x + .5\min\{x, y\} > .5\}$ and $\{(x, y) \in [0, 1]^2 \mid \min\{x, y\} > .5\}$ respectively.

Figure 5.1 shows the flexibility of the Choquet integral facing binary class classification problem.

5.1.2 Dual Form

The basic idea of kernel-based learning and the dual form setting have been presented in Subsection 2.4.3. The main idea here is to embed the Choquet integral in a kernel-based framework. To this end, the so-called Choquet kernel comes into play. Basically the core idea is to reduce the complexity of the problem (from 2^m to $m^2 \log m$) by defining a new kernel. In this case, we seek the $\{\alpha_i\}_{i=1}^n$, which minimize the following objective function under constraints.

$$\min_{\boldsymbol{\alpha}} \left\{ \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K_{\mathcal{C}}^{k=p}(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i=1}^{n} \alpha_i \right\}$$

s.t.

$$\sum_{i=1}^{n} y_i \alpha_i = 0$$

$$0 \leq \alpha_i \leq C \qquad \forall i \in \{1, \ldots, n\}$$

Here C is a typical SVM trade-off parameter and $K_{\mathcal{C}}^{k=p}$ is referred to the Choquet kernel of degree p. In this new setting, so far we did not consider the constraints in (5.1). At this point, it should be clarified, that the setting underlying the SVM is quite different compared to the setting underlying the kernels method in terms of

ensuring monotonicity constraints. In SVM-setting all monotonicity constraints are taken into consideration, whereas in kernel-based learning there is no monotonicity constraint at all. Roughly speaking, in this case, we do not want to enforce the monotonicity constraints artificially in our setting. Accounting for the fact that, satisfying monotonicity constraints can be done partially by training examples, the hope is, due to the monotonicity property of training example (of course depends on the training examples), such a property can be inherited by the the model-parameters at least partially.

Such expectations are not at all unrealistic, and in the following we show how empirically those expectations can be satisfied. It should be once again emphasized that if the training examples hold monotonicity property related to input and output spaces, this kind of properties can be captured by the model inheritably. In general, the core idea of learning is to bias the observations, although such expectations can be satisfied partially.

Main Motivation

As discussed earlier, the complexity of learning the Choquet integral, more precisely estimating parameters for the Möbius transform, even for small k- additive case is expensive. Actually what makes the learning process more sophisticated, are the monotonicity constraints; which so far always had to be considered. Loosely speaking, one also can think on learning the Choquet integral without any monotonicity constraints (relax optimization) and after estimating the parameters try to fix/correct the inconsistencies. Basically the expectation is that during the learning process, the monotone data can enforce monotonicity behavior (at least partially) to our model. Of course we cannot expect the extracted model to be quite monotone, but at least there is a hope to satisfy monotonicity constraints partially. Finally, the algorithm fixes the inconsistencies in the fuzzy measure. To this end, several algorithms to fix the inconsistencies are presented in Chapter 6.

5.2 The Choquet Kernels

In this section we consider solely the feature mapping built by basis functions in equation (3.3). More precisely, the expression in (3.3) can also be written in terms of an inner product

$$\Big\langle \boldsymbol{m}_{\mathfrak{Z}}, \varphi(f(\boldsymbol{x})) \Big\rangle \ ,$$

with the mapping $\varphi : \mathbb{R}^m \to \mathbb{R}^{2^m-1}$ defined as follows:

$$\varphi(\boldsymbol{x}) = \varphi(x_1, \ldots, x_m) = \Big(x_1, \ldots, x_m, \min\{x_1, x_2\}, \ldots, \min\{x_{m-1}, x_m\}, \\ \min\{x_1, x_2, x_3\}, \ldots, \min\{x_1, \ldots, x_m\}\Big)$$

and in addition $\boldsymbol{m}_\mathfrak{I}$ denotes the vector

$$\Big(\boldsymbol{m}(\{c_1\}), \ldots, \boldsymbol{m}(\{c_n\}), \boldsymbol{m}(\{c_1, c_2\}), \ldots, \boldsymbol{m}(\{c_{n-1}, c_n\}), \ldots, \boldsymbol{m}(\{c_1, \ldots, c_n\})\Big).$$

Let us investigate more in the details feature mapping $\varphi(\cdot)$. Basically $\varphi(\cdot)$ can be seen as a mapping, which maps the elements from $\mathbb{R}^m$ to $\mathbb{R}^{2^m-1}$. Assuming $\boldsymbol{x}, \boldsymbol{x}^* \in \mathbb{R}^m$, this notation allows us to define an inner product between $\varphi(\boldsymbol{x})$ and $\varphi(\boldsymbol{x^*})$ as follows:

$$\Big\langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x^*}) \Big\rangle = x_1 x_1^* + \ldots + x_m x_m^* + \min\{x_1, x_2\} \min\{x_1^*, x_2^*\} + \ldots \\ + \min\{x_1, x_2, \ldots, x_m\} \min\{x_1^*, x_2^*, \ldots, x_m^*\}$$

We write $\Big\langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x^*}) \Big\rangle$ as $K_\mathcal{C}^{k=m}(\boldsymbol{x}, \boldsymbol{x}^*)$, and claim that $K_\mathcal{C}^{k=m}(\boldsymbol{x}, \boldsymbol{x}^*)$ is a kernel. This claim is validated for the well-known kernel properties; it is defined on the inner product of feature mapping $\varphi(\cdot)$.

So far $K_\mathcal{C}^{k=m}(\boldsymbol{x}, \boldsymbol{x}^*)$ has a computational complexity of $\mathcal{O}(2^m)$, because of $2^m - 1$ summands. Our idea refers to computing the kernel, which we call it henceforth the *Choquet kernel*, in the most efficient way. Loosely speaking, the idea is to compute $K_\mathcal{C}^{k=m}$ in a way that the computational complexity is reduced to $\mathcal{O}(m^2)$.

Full Choquet Kernel

Proposition 5.1 *$K_\mathcal{C}^{k=m}(\boldsymbol{x}, \boldsymbol{x}^*)$ can be reformulated as follows:*

$$K_\mathcal{C}^{k=m}(\boldsymbol{x}, \boldsymbol{x}^*) = \Big\langle \boldsymbol{x}, \boldsymbol{x}^* \Big\rangle + \sum_{i=1}^{m-1} x_{\sigma_i} \cdot \left\{ \sum_{j=0}^{m-1-i} 2^{m-1-i-j} \min\left\{ x_{\sigma_i}^*, x_{\Psi_{i,j+1}^*}^* \right\} \right\} \quad ,$$

where σ sorts the values of first instance in increasing way, namely, $x_{\sigma_1} \leq \ldots \leq x_{\sigma_m}$, namely

$$\begin{array}{ccc} x_{\sigma_1} \leq \ldots \leq x_{\sigma_m} \\ \downarrow \quad \ldots\ldots \quad \downarrow \\ x_{\sigma_1}^* \quad \ldots\ldots\ldots \quad x_{\sigma_m}^* \end{array}$$

and $x^*_{\Psi^*_{i,l}}$ *indicates the* l*-th ordered value in subset* $\left\{x^*_{\sigma_i}, ..., x^*_{\sigma_m}\right\}$.

Proof 5.1 *We start first with* $\left\langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}^*) \right\rangle$. *With out loss of generality assume that there are two permutations, namely* σ*, where*

$$\begin{array}{ccc} x_{\sigma_1} & \leq \ldots \leq & x_{\sigma_m} \\ \downarrow & \ldots\ldots & \downarrow \\ x^*_{\sigma_1} & \ldots\ldots & x^*_{\sigma_m} \end{array}$$

In general, $\left\langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}^*) \right\rangle$ *can be written as follows:*

$$\begin{aligned} \left\langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}^*) \right\rangle &= \left\langle \varphi(\sigma(\boldsymbol{x})), \varphi(\sigma(\boldsymbol{x}^*)) \right\rangle \\ &= x_{\sigma_1} x^*_{\sigma_1} + \ldots + x_{\sigma_m} x^*_{\sigma_m} + \min\{x_{\sigma_1}, x_{\sigma_2}\} \min\{x^*_{\sigma_1}, x^*_{\sigma_2}\} \\ &+ \ldots + \min\{x_{\sigma_1}, x_{\sigma_2}, \ldots, x_{\sigma_m}\} \min\{x^*_{\sigma_1}, x^*_{\sigma_2}, \ldots, x^*_{\sigma_m}\} \qquad (5.2) \\ &= \left\langle \boldsymbol{x}, \boldsymbol{x}^* \right\rangle + \sum_{i=1}^{m-1} x_{\sigma_i} \cdot \left\{ \sum_{s=1}^{m-i} \wp(i,s) \right\}, \end{aligned}$$

where additionally

$$\begin{aligned} \wp(i,s) &= \sum_{i<j<k<\ldots<p\leq m} \min\left\{x^*_{\sigma_i}, x^*_{\sigma_j}, x^*_{\sigma_k}, \ldots, x^*_{\sigma_p}\right\} \\ &= \underbrace{\sum_{j=i+1}^{m} \sum_{k=j+1}^{m} \cdots \sum_{p=o+1}^{m}}_{s\text{-summations}} \min\left\{x^*_{\sigma_i}, x^*_{\sigma_j}, x^*_{\sigma_k}, \ldots, x^*_{\sigma_p}\right\} \end{aligned}$$

It is also easy to check that when ξ *is a permutation such that*

$$x_{\xi_1} \leq \ldots \leq x_{\xi_p}$$

then

$$\sum_{T\subseteq\{x_{\xi_1},\ldots,x_{\xi_p}\}} \min\{T\} = \sum_{i=0}^{p-1} 2^{p-1-i} x_{\xi_{i+1}} \qquad (5.3)$$

Note that, here $1 \leq |T| \leq p$. *Let us assume that given instance* $\boldsymbol{x}^* = (x^*_1, \ldots, x^*_p)$, $x^*_{\Psi^*_j}$ *is* j*-th ordered value among* $\{x^*_1, \ldots, x^*_p\}$. *Moreover assume* $x^*_{\Delta^*}$ *is a different*

value for which we do not know the order of this value among $\{x_1^*, \ldots, x_p^*\}$*. The goal is to compute* $\sum_{T \subseteq \{x_1^*, \ldots, x_p^*\}} \min\{x_\sigma^*, T\}$*. Since we do not know the position of* x_σ^* *we use the following trick:*

$$\min\left\{x_\sigma^*, T\right\} = \min\left\{x_\sigma^*, \min\{T\}\right\}$$

Using equation 5.3 and the previous line, we get the following equation:

$$\sum_{T \subseteq \{x_1^*, \ldots, x_p^*\}} \min\{x_\sigma^*, T\} = \sum_{j=0}^{p-1} 2^{p-1-j} \min\left\{x_\sigma, x_{\Psi_j^*}\right\} \ ,$$

where $x_{\Psi_j^*}^*$ *is* j*-th ordered value among* $\{x_1^*, \ldots, x_p^*\}$*. In other words*

$$\sum_{T \subseteq \{x_{\xi_k^*} | k > r\}} \min\{x_\sigma, x_{\xi_r^*}, T\} = 2^{|T|} \min\{x_\sigma, x_{\xi_r^*}\} \ ,$$

where $T = \emptyset$ *is as well considered and*

$$x_{\xi_1^*}^* \leq \ldots \leq x_{\xi_p^*}^* \ .$$

Then

$$\sum_{s=1}^{m-i} \wp(i, s) = \sum_{j=0}^{m-1-i} 2^{m-1-i-j} \min\left\{x_{\sigma_i}, x_{\Psi_{i,j+1}^*}\right\} \ .$$

Substituting the above equation in (5.2), we get the compact form of the Choquet kernel. ■

For instance assume $\boldsymbol{x} = (2, 1, 3), \boldsymbol{x}^* = (1, 9, 7)$. Then $\left\langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}^*) \right\rangle = (2{+}9{+}21){+}(1{+}2{+}7{+}1) = 43$. Based on our definition $\sigma(\boldsymbol{x}) = (1, 2, 3), \sigma(\boldsymbol{x}^*) = (9, 1, 7)$ and finally $K_{\mathcal{C}}^{k=3}(\boldsymbol{x}, \boldsymbol{x}^*) = (2{+}9{+}21){+}1{\times}\{2^1{\times}1{+}2^0{\times}7\}{+}2{\times}\{2^0{\times}1\} = 43$.

Remark 5.1 ***(complexity Issue)*** *In order to show that the complexity of the Choquet kernel is quadratic, first of all the elements of* $\boldsymbol{x}$ *are sorted. Assume the elements of* $\boldsymbol{x}$ *are already sorted by permutation* σ*. Then suppose the Choquet kernel as follows:*

$$\sum_{i=1}^{m-1} x_{\sigma_i} \cdot \left\{ \sum_{j=0}^{m-1-i} 2^{m-1-i-j} \min \left\{ x^*_{\sigma_i}, x^*_{\Psi^*_{i,j+1}} \right\} \right\} \ .$$

The inner summation has complexity of $\mathcal{O}(m)$ and since there are two summands the whole complexity is $\mathcal{O}(m^2)$. Moreover, the complexity of sorting the elements is $\mathcal{O}(m \log m)$. Considering these facts together, in the end the computational complexity of full Choquet kernel is $\mathcal{O}(m^2 + m \log m) = \mathcal{O}(m^2)$.

2-additive Choquet Kernel

Corollary 5.1 *For $k = 2$ the Choquet kernel can be reformulated as follows:*

$$K_{\mathcal{C}}^{k=2}(\boldsymbol{x}, \boldsymbol{x}^*) = \left\langle X, X^* \right\rangle + \sum_{i=1}^{m-1} x_{\sigma_i} \cdot \left\{ \sum_{j=i}^{m} \min \left\{ x^*_{\sigma_i}, x^*_{\sigma_j} \right\} \right\} \ .$$

3-additive Choquet Kernel

Corollary 5.2 *For $k = 3$ the Choquet kernel can be reformulated as follows:*

$$K_{\mathcal{C}}^{k=3}(\boldsymbol{x}, \boldsymbol{x}^*) = \left\langle X, X^* \right\rangle + \sum_{i=1}^{m-1} x_{\sigma_i} \cdot \left\{ \sum_{j=1}^{m-i} (m-i-j) \min \left\{ x^*_{\sigma_i}, x^*_{\Psi^*_{i,j}} \right\} \right\} \ .$$

Proof 5.2 *The idea of proof is the same as proposition 1, with the following equation:*

$$\sum_{\substack{T \subseteq \{x_1, \ldots, x_p\} \\ |T| < 3}} \min \left\{ x^*_{\sigma_i}, T \right\} = \sum_{k=0}^{p} (p-k) \min \left\{ x^*_{\sigma_i}, x^*_{\Psi^*_k} \right\} \ .$$

■

4-additive Choquet Kernel

Corollary 5.3 *For $k = 4$ the Choquet kernel can be reformulated as follows:*

$$K_{\mathcal{C}}^{k=4}(\boldsymbol{x}, \boldsymbol{x}^*) = \left\langle \boldsymbol{x}, \boldsymbol{x}^* \right\rangle + \sum_{i=1}^{m-1} x_{\sigma_i} \cdot \left\{ \sum_{j=0}^{m-i-1} \left\{ \binom{m-i+1-j}{2} + 1 \right\} \min \left\{ x^*_{\sigma_i}, x^*_{\Psi^*_{i,j+1}} \right\} \right\} \ .$$

Proof 5.3 *The idea of proof is the same as proposition 1, with the following equation:*

$$\sum_{\substack{T\subseteq\{x_1,\ldots,x_p\}\\|T|<4}} \min\left\{x^*_{\sigma_i}, T\right\} = \sum_{k=1}^{p-1}\left\{\binom{p+1-k}{2}+1\right\}\min\left\{x^*_{\sigma_i}, x^*_{\Psi^*_k}\right\} \ .$$

■

6
Capacity Control

So far two different inductive principles were taken into consideration to establish monotone classifiers based on the powerful aggregation function, namely the Choquet integral. From a machine learning point of view, of course a natural question is, how flexible are the proposed classifiers? This question is addressed as the so-called VC dimension. This chapter begins by exploiting some theoretical results regarding the VC dimension of the Choquet integral. The theoretical results confirm that the VC dimension of the model class regarding of the Choquet integral is high. As discussed earlier, the high degree of flexibility exposes several disadvantages, e.g., the overfitting problem. As can be seen, there is indeed a demand to reduce this flexibility with respect to different training examples. To this end, the typical approach is to regularize the model, which is addressed as regularization methods. Considering the overfitting issue, we propose two types of regularization and we introduce the algorithms in Section 6.2.

Since the Choquet integral generally for full case and particularly for $k-$additive case has an exponential number of constraints to ensure monotonicity, learning this kind of model is indeed highly complex. Therefore, one common question is how can the related complexity be reduced? In particular, how is it possible to reduce the number of monotonicity constraints? For this purpose, in this chapter several algorithms for the complexity reduction issue are presented. Regarding the complexity issue, there are many types of measurement to quantify the complexity, e.g.,

in terms of running time or in terms of capacity. For instance, two algorithms can have the same computational complexity from a theoretical point of view, but in practice they are different from a running time point of view. In this view, we take into consideration both aspects and the related results are presented.

The reduction of computational complexity is usually considered for supervised learning. The computational complexity reduction also can be investigated under unsupervised learning. In particular, in the Choquet integral case, since in the feature space there are an exponential number of features, it is quite necessary to exploit some dependencies in the feature space. This kind of reduction is addressed in Section 6.3.1.

As mentioned, the level of complexity of the Choquet integral can be chosen under k-additivity (3.3.4) in advance. In this Chapter, the 2-additive case is considered, and two different computational complexity reductions are shown. In fact, there are several reasons to consider the 2-additive case. The 2-additive case is the first non-linear level complexity from the Choquet integral. In the 2-additive case, solely the interactions in a pairwise manner are taken into account, which are more interpretable and comprehensive. Note that, although in the 2-additive case, only the pairs are considered, the number of constraints to ensure monotonicity are exponential. It is shown in Subsection 6.3.2 that the exponential complexity can be reduced to the quadratic complexity.

So far, we always considered the monotonicity constraints during learning process. In general, one may also think about learning the weight parameters without any monotonicity constraints. The expectation is that the constraints can be satisfied at least partially. In Section 6.4 we present the core idea to fix the inconsistencies among the weights for fuzzy measure. Remember that the ultimate goal in this thesis is to estimate the proper fuzzy measure given some observations. As mentioned earlier, the goal is to learn the Choquet integral as an aggregation function in our approaches. Loosely speaking, the learning problem comes down to learning the fuzzy measure. In Section 6.4 the core idea to monotonize the non-monotone measure is discussed. Before continuing the topic, it should be emphasized that by *capacity* we mean the flexibility of a classifier and should not be confused with the concept of capacity of a measure in Chapter 3.
Parts of this chapter were already published in [59, 60, 98, 101].

6.1 Under VC Dimension of the Choquet Integral

The basic definition and idea of the VC dimension has been discussed in Subsection 2.3.2. In this section, we discuss the VC dimension of the Choquet integral. Advocating the Choquet integral as a novel tool for machine learning immediately begs an interesting theoretical question, namely the question regarding the capacity of the corresponding model class. In fact, since the Choquet integral in its general form (not restricted to k-additive measures) has a rather large number of parameters, one may expect it to be quite flexible and, therefore, to have a high capacity. On the other hand, the parameters cannot be chosen freely. Instead, they are highly constrained due to the properties of the underlying fuzzy measure.

In this section, we are going to analyze the capacity of the Choquet integral in terms of the VC dimension [108]. To this end, we consider a setting in which the Choquet integral is used to classify instances represented in the form of m-dimensional vectors $\boldsymbol{x} = (x_1, x_2, \ldots, x_m) \in \mathbb{R}_+^m$, where $x_i = f(c_i)$ can be thought of as the evaluation of the criterion c_i. More specifically, we consider the model class $\mathcal{H}$ consisting of all threshold classifiers of the form

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_m) \mapsto \mathbb{I}\Big(\mathcal{C}_\mu(\boldsymbol{x}) > \beta\Big) \ , \tag{6.1}$$

where $\mathbb{I}$ maps truth degrees $\{\text{false}, \text{true}\}$ to $\{0, 1\}$ and as usual, μ is the fuzzy measure, $\mathcal{C}_\mu(\boldsymbol{x})$ is the Choquet integral of the (normalized) attribute values $x_1, x_2, \ldots, x_m$ and $\beta \in [0, 1]$ is a threshold value (since this part is responsible for the classification decision, results on the VC dimension of $\mathcal{H}$ directly apply to the choquistic regression, as well). Note that the class $\mathcal{H}$ is parametrized by μ and β.

Theorem 6.1 *For the model class $\mathcal{H}$ as defined above, $VC(\mathcal{H}) = \Omega(2^m/\sqrt{m})$. That is, the VC dimension of $\mathcal{H}$ grows asymptotically at least as fast as $2^m/\sqrt{m}$.*

Proof 6.1 *In order to prove this claim, we construct a sufficiently large data set $\mathcal{D}$ and show that, despite its size, it can be shattered by $\mathcal{H}$. In this construction, we restrict ourselves to binary attribute values, which means that $x_i \in \{0, 1\}$ for all $1 \leq i \leq m$. Consequently, each instance $\boldsymbol{x} = (x_1, \ldots, x_m) \in \{0, 1\}^m$ can be identified with a subset of indices $S_{\boldsymbol{x}} \subseteq X = \{1, 2, \ldots, m\}$, namely its* indicator set *$S_{\boldsymbol{x}} = \{i \,|\, x_i = 1\}$.*

In combinatorics, an antichain *of $X = \{1, 2, \ldots, m\}$ is a family of subsets $\mathcal{A} \subset 2^X$ such that, for all $A, B \in \mathcal{A}$, neither $A \subseteq B$ nor $B \subseteq A$. An interesting question related to the notion of an antichain concerns its potential size, that is,*

the number of subsets in $\mathcal{A}$. This number is obviously restricted due to the above non-inclusion constraint on pairs of subsets. An answer to this question is given by a well-known result of Sperner [92], who showed that this number is

$$\binom{m}{\lfloor m/2 \rfloor}. \tag{6.2}$$

Moreover, Sperner has shown that the corresponding antichain $\mathcal{A}$ is given by the family of all q-subsets of X with $q = \lfloor m/2 \rfloor$, that is, all subsets $A \subset X$ such that $|A| = q$.

Now, we define the data set $\mathcal{D}$ in terms of the collection of all instances $\boldsymbol{x} = (x_1, \ldots, x_m) \in \{0,1\}^m$ whose indicator set $S_{\boldsymbol{x}}$ is a q-subset of X. Recall that, from a decision making perspective, each attribute can be interpreted as a criterion. Thus, each instance in our data set satisfies exactly q of the m criteria, and there is not a single "dominance" relation in the sense that the set of criteria satisfied by one instance is a superset of those satisfied by another instance. Intuitively, the instances in $\mathcal{D}$ are therefore maximally incomparable. This is precisely the property we are now going to exploit in order to show that $\mathcal{D}$ can be shattered by $\mathcal{H}$.

Recall that a set of instances $\mathcal{D}$ can be shattered by a model class $\mathcal{H}$ if, for each subset $\mathcal{P} \subseteq \mathcal{D}$, there is a model $H \in \mathcal{H}$ such that $H(\boldsymbol{x}) = 1$ for all $\boldsymbol{x} \in \mathcal{P}$ and $H(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathcal{D} \setminus \mathcal{P}$. Now, take any such subset $\mathcal{P}$ from our data set $\mathcal{D}$ as constructed above, and recall that the Choquet integral in (6.1) can be written as

$$\mathcal{C}_\mu(\boldsymbol{x}) = \sum_{T \subseteq C} \boldsymbol{m}(T) \times f_T(\boldsymbol{x}) \ , \tag{6.3}$$

where $f_T(\boldsymbol{x}) = 1$ if $T \subseteq S_{\boldsymbol{x}}$ and $f_T(\boldsymbol{x}) = 0$ otherwise. We define the values $\boldsymbol{m}(T)$, $T \subseteq C$, of the Möbius transform as follows:

$$\boldsymbol{m}(T) = \begin{cases} |\mathcal{P}|^{-1} & \text{if } T = S_{\boldsymbol{x}} \text{ for some } \boldsymbol{x} \in \mathcal{P} \\ 0 & \text{otherwise} \end{cases} .$$

Obviously, this definition of the Möbius transform is feasible and yields a proper fuzzy measure μ: The sum of masses is equal to 1, and since all masses are non-negative, monotonicity is ensured right away. Moreover, from the construction of $\boldsymbol{m}$ and the fact that, for each pair $\boldsymbol{x} \neq \boldsymbol{x}' \in \mathcal{D}$, neither $S_{\boldsymbol{x}} \subseteq S_{\boldsymbol{x}'}$ nor $S_{\boldsymbol{x}'} \subseteq S_{\boldsymbol{x}}$, the Choquet integral is obviously given as follows:

$$\mathcal{C}_\mu = \begin{cases} |\mathcal{P}|^{-1} & \textit{if } \boldsymbol{x} \in \mathcal{P} \\ 0 & \textit{otherwise} \end{cases} .$$

Thus with $\beta = 1/(2|\mathcal{P}|)$, the classifier (6.1) behaves exactly as required, that is, it classifies all $\boldsymbol{x} \in \mathcal{P}$ as positive and all $\boldsymbol{x} \notin \mathcal{P}$ as negative.

Noting that the special case where $\mathcal{P} = \emptyset$ is handled correctly by the Möbius transform $\boldsymbol{m}$ such that $\boldsymbol{m}(C) = 1$ and $\boldsymbol{m}(T) = 0$ for all $T \subsetneq C$ (and any threshold $\beta > 0$), we can conclude that the data set $\mathcal{D}$ can be shattered by $\mathcal{H}$. Consequently, the VC dimension of $\mathcal{H}$ is at least the size of $\mathcal{D}$, whence (6.2) is a lower bound of $VC(\mathcal{H})$.

For the asymptotic analysis, we make use of Sterling's approximation of large factorials (and hence binomial coefficients). For the sequence $(b_1, b_2, \ldots)$ of the so-called central binomial coefficients b_n, it is known that

$$b_n = \binom{2n}{n} = \frac{(2n)!}{(n!)^2} \geq \frac{1}{2}\frac{4^n}{\sqrt{\pi \cdot n}} .$$

Thus, the fact that $VC(\mathcal{H})$ grows asymptotically at least as fast as $2^m/\sqrt{m}$ immediately follows by setting $n = m/2$ and ignoring constant terms.

Remark 6.2 *Recall the expression (3.3) of the Choquet integral in terms of its Möbius transform. This expression shows that the Choquet integral corresponds to a linear function, albeit a constrained one, in the* feature space *spanned by the set of features $\{f_T \,|\, T \subseteq \{1, 2, \ldots, m\}\}$ (already used in (6.3)) , where each feature is a min-term*

$$f_T = f_T(\boldsymbol{x}) = f_T(x_1, \ldots, x_m) = \min_{i \in T} x_i \ . \tag{6.4}$$

The dimensionality of this feature space is $2^m - 1$. Thus, it follows immediately that $VC(\mathcal{H}) \leq 2^m$ (the class of linear hyperplanes in $\mathbb{R}^m$ has VC dimension $m + 1$). Together with the lower bound $2^m/\sqrt{m}$, which is not much smaller (despite the restriction to binary attribute vectors), we thus dispose of a relatively tight approximation of $VC(\mathcal{H})$.

Remark 6.3 *Interestingly, the proof of Theorem 6.1 does not exploit the full non-additivity of the Choquet integral. In fact, the measure we constructed there is $\lfloor m/2 \rfloor$-additive, since $\boldsymbol{m}(T) = 0$ for all $T \subseteq C$ with $|T| > \lfloor m/2 \rfloor$. Consequently,*

the estimation of the VC dimension still applies to the restricted case of k-additive measures, provided $k \geq \lfloor m/2 \rfloor$. For smaller k, it is not difficult to adapt the proof so as to show that

$$VC(\mathcal{H}) \geq \binom{m}{k} . \tag{6.5}$$

In this regard, it is also quite interesting to compare the above results with practical results in [13, 84].

6.2 Regularization

One of the common problems, which can occur during the learning process is the overfitting problem. The problem usually occurs when the number of attributes (in the feature space of the model) compared to the number of samples are relatively high. In order to overcome this problem, so far we have considered the reduction in terms of model-restriction. Specifically, given a k, k-additive Choquet integral and particularly the $k = 2$, 2-additive Choquet integral, has been chosen in advance. This means, we restricted our model to a certain degree of interaction. More precisely, considering a k-additive model, automatically implies that the l-way interactions, for $l > k$, are not taken into account. In fact, selecting the model in advance causes to ignore some interactions unwillingly.

There is also another way to overcome the over-fitting problem. From a machine learning point of view, typically the regularization technique is used to prevent overfitting. Roughly speaking, the regularization technique is to add artificially more constraints on parameters in order to restrict the model in a sound way. The selection or design of the constraints are mainly related to the structure of model. For instance, while a specific regularization for a certain model can improve the performance, it can decline the performance of other models. In this section, two types of regularizations are presented and discussed in the more detail the properties of each regularization.

6.2.1 L_1-Regularization

From a machine learning point of view, one traditional way to prevent overfitting is to use L_1-regularization. Basically L_1-regularization is added to an objective function as the following form:

$$\min(\max)\Big\{OF(\boldsymbol{\nu})\Big\} + \min(-\max)\Big\{\sum_{i=1}^{p}|\nu_i|\Big\} \ ,$$

where $\boldsymbol{\nu} = \big\{\nu_i\big\}_{i=1}^{p}$ are the model parameters and $OF(\cdot)$ is assigned to the objective function. As can be seen, the L_1-regularization prevents unnecessary weights.

For our setting, namely, the choquistic regression we proposed the following form:

$$\begin{aligned}\max_{\boldsymbol{m},\gamma,\beta} \Big\{ & -\gamma \sum_{i=1}^{n}(1-y^{(i)})\Big(\mathcal{C}_{\boldsymbol{m}}(\boldsymbol{x}^{(i)}) - \beta\Big) \\ & - \sum_{i=1}^{n}\log\big(1+\exp(-\gamma\left(\mathcal{C}_{\boldsymbol{m}}(\boldsymbol{x}^{(i)}) - \beta\right))\big) - \eta\sum_{T\subseteq C}|\boldsymbol{m}(T)|\Big\}\end{aligned} \tag{6.6}$$

s.t.

$$\eta, \gamma > 0, \ 0 \leq \beta \leq 1$$

$$\sum_{T\subseteq C}\boldsymbol{m}(T) = 1,$$

$$\sum_{B\subseteq A\setminus\{c_i\}}\boldsymbol{m}(B\cup\{c_i\}) \geq 0 \quad \forall A \subseteq C, \forall c_i \in C.$$

The last part of the objective function (6.6) is a standard L_1-regularizer on the Möbius transform, which is added as a means to prevent over-fitting; moreover, since many weights are typically set to 0 under L_1-regularization, it also serves as a feature selection mechanism [70]. Of course this idea also can be applied to other learning problem. The corresponding results for choquistic regression are shown in Chapter 7 Table 7.3.

6.2.2 Hierarchical Regularization

As mentioned previously, the concept of regularization is absolutely related to the model class assumption. In this section we proposed a specific kind of regularization, which takes to the consideration different levels of complexity. Indeed, the idea is to use different regularization terms for different levels of complexity. Roughly speaking the core idea is to weight different parameters from different levels by different weights. More precisely, we make a distinction between parameters

when the parameters do not belong to same level of k-additivity. More formally, the regularization term is formalized as follows:

$$l = \gamma\Big(\sum_{A \subseteq C} f(|A|)\big|\boldsymbol{m}(A)\big|\Big) \quad , \tag{6.7}$$

where $f(\cdot)$ is a strictly increasing function. Defining $f(\cdot)$ as a strictly increasing function, this term encourages $\boldsymbol{m}(A) = 0$ for larger subsets of criteria A; in other words, it encourages a restriction to measures with a low level of non-additivity. We note that (6.7) can be seen as a specific instance of the idea of "hierarchical regularization" as introduced in [5], with a hierarchy on the power set 2^C defined through subset cardinality (i.e., the first level of the hierarchy are the singletons $\{c_i\}$, the second level the two-subsets $\{c_i, c_j\}$, etc.).

Also the magnitude of γ and f indicate the complexity of the model. The higher the γ and larger the f, the less complex the model, whereas the lower the γ and the smaller the f, the more complex the model.

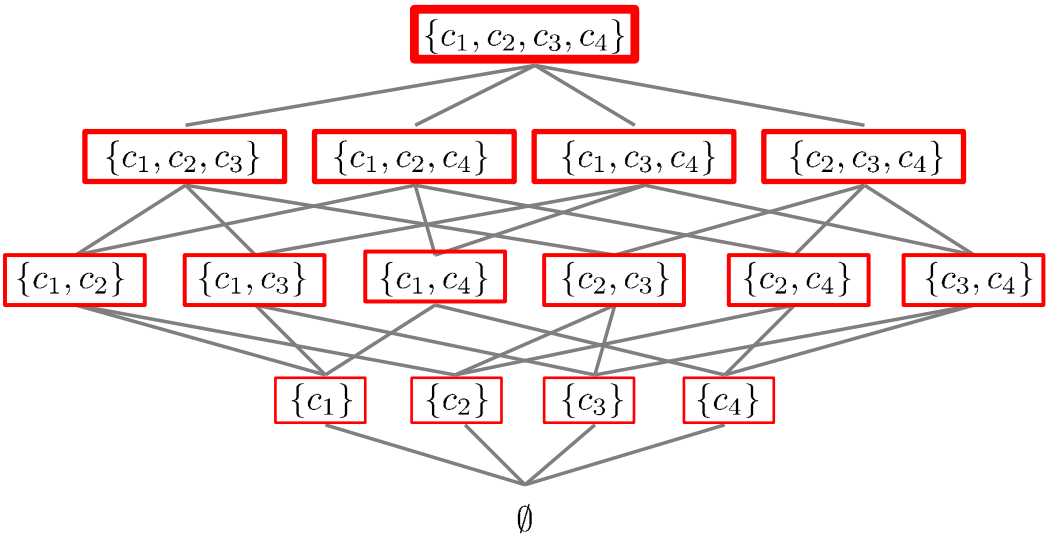

Figure 6.1: Illustrative of Hierarchical Regularization for 4 criteria; the dicker the length of box, the more penalization of the joint weights

The corresponding results are shown in Chapter 7 Table 7.10 as OCR+R.

6.3 Complexity Reduction

6.3.1 Complexity Reduction by Exploiting Dependency

Obviously, the Choquet integral can be interpreted as fitting a (constrained) linear function in the feature space spanned by the set of features f_T defined by (6.4), with one feature for each subset of criteria $T \subseteq \{c_1, c_2, \ldots, c_m\}$. Since the dimensionality of this feature space is $2^m - 1$, the method is clearly critical from a complexity point of view. To reduce this complexity in we propose in this section an idea which takes into consideration the data. In other words, the data can be used as a hint to reduce the complexity.

In this regard, one may wonder whether some of the features from (6.4) could not even be eliminated *prior* to solving the actual optimization problem. Specifically interesting in this regard is a possible restriction of the Choquet integral to k-additive measures, for a suitable value of $k < m$. Besides, a restriction to k-additive measures may also have advantages from a learning point of view, as it reduces the capacity of the underlying model class and thus may prevent over-fitting the data in cases where the full flexibility of the Choquet integral is actually not needed. Of course, the key problem to be addressed concerns the question of how to choose k in the most favorable way.

Exploiting Equivalence of Features for Dimensionality Reduction

In the following, we shall elaborate on the following question: Is it possible to find an upper bound on the required level of complexity of the model, namely the level of additivity k, prior to fitting the Choquet integral to the data? Or, more specifically, can we determine the value k in such a way that fitting a k-additive measure is definitely enough, in the sense that each labeling of the training data produced by the full Choquet integral ($k = m$) can also be produced by a Choquet integral based on a k-additive measure?

In this regard, it is noticeable that, for a given instance $\boldsymbol{x} = (x_1, \ldots, x_m)$, many of the min-terms (6.4) will assume the same value (in fact, there are $2^m - 1$ such terms but only m possible values). Consequently, in the expression

$$\mathcal{C}_\mu(\boldsymbol{x}) = \sum_{T \subseteq C} \boldsymbol{m}(T) \times f_T(\boldsymbol{x}) \tag{6.8}$$

of the Choquet integral, many coefficients $\boldsymbol{m}(T)$ can be grouped and, in principle, be replaced by a single one. The groups thus defined solely depend on the order of

the values $x_1, \ldots, x_m$ of the original attributes. The number of terms in (6.8) will thus reduce from $2^m - 1$ to at most m. However, since the order may change from instance to instance, different groupings may be obtained for different instances.

Now, imagine that a subset of features $\mathcal{F} = \{f_{T_1}, \ldots, f_{T_r}\}$ assumes the same value, not only for a single instance, but for all instances in the training data. Then, this set can be said to form an equivalence class. Thus, one of the features could in principle be selected as a representative, absorbing all the weights of the others; more specifically, the weight of this feature would be set to $\boldsymbol{m}(T_1) + \boldsymbol{m}(T_2) + \ldots + \boldsymbol{m}(T_r)$, while the weights of the other features in $\mathcal{F}$ would be set to 0.

Note, however, that this "transfer of Möbius mass" will in general not be feasible, as it may cause a violation of the monotonicity constraint on the fuzzy measure μ. As a side remark, we also note that, from a learning point of view, the equivalence of features may obviously cause problems with regard to the identifiability of coefficients; due to the monotonicity constraints just mentioned, however, this is not necessarily the case.

More generally, for two features f_A and f_B ($A, B \subseteq C$), denote by $v(A, B) \in [0, 1]$ the fraction of training examples on which they assume the same value. We say that f_A covers f_B (and, vice versa, f_B covers f_A) if $v(A, B) = 1$. Moreover, for a feature f_A, we denote by $C(f_A) \subseteq 2^C$ the set of features it covers. A straightforward way to find a sufficiently large k then consists of finding the smallest k such that

$$\bigcup_{T \subseteq C, |T| \leq k} C(f_T) = 2^C \quad . \tag{6.9}$$

From the above construction, it follows that working with the corresponding k-additive measure, for k thus defined, is theoretically sound and guarantees that there is no loss in terms of the expressivity of the model on the training data. We summarize this finding in terms of the following proposition.

Proposition 6.1 *Consider a set of training instances $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}$ and let k^* be the smallest value in $\{1, \ldots, m\}$ satisfying (6.9). Moreover, let μ be any measure on the set of criteria $\{c_1, \ldots, c_m\}$, and $\mathcal{C}_\mu$ the Choquet integral with respect to this measure. Then, there exists a k-additive measure μ^* such that*

$$\mathcal{C}_{\mu^*}(\boldsymbol{x}^{(i)}) = \mathcal{C}_\mu(\boldsymbol{x}^{(i)}) \tag{6.10}$$

for all $i \in \{1, \ldots, m\}$.

We like to emphasize that k^* is only an upper bound on the complexity needed to fit the training data. Thus, it is not necessarily the optimal k from the point of

view of model induction (which might be figured out by the regularizer in (6.6)). In particular, note that the computation of k^* does not refer to the output values $y^{(i)}$. Instead, it should be considered as a measure of the complexity of the training instances. As such, it is obviously connected to the notion of the VC dimension.

Since the exact reproducibility (6.10) may appear overly stringent or, stated differently, a small loss may actually be acceptable, we finally propose a relaxation somewhat in line with the idea of *probably approximately correct* (PAC) learning [106]. First, noting that the Choquet integral may change by at most ϵ when combining features f_A and f_B such that $|f_A - f_B| < \epsilon$, one may think of relaxing the definition of equivalence as follows: f_A and f_B are ϵ-equivalent (on a given training instance $\boldsymbol{x}$) if $|f_A(\boldsymbol{x}) - f_B(\boldsymbol{x})| < \epsilon$. Second, we relax the condition of coverage. Denoting by $v(A, B) \in [0, 1]$ the fraction of training examples on which f_A and f_B are ϵ-equivalent, we say that f_A ϵ-δ-covers f_B if $v(A, B) \geq 1 - \delta$.

Algorithm 1 k^* - algorithm

Inputs: ϵ, δ, $C = \{c_1, \ldots, c_m\}$, training data: $TR = \{\boldsymbol{x}_i\}_{i=1}^n$ and features $\mathcal{F} = \{f_T \mid T \subseteq C\}$

$k = |C|$
while $k > 1$ **do**
  $G_k = \{f_T \mid |T| = k\}$
  for $f_T \in G_k$ **do**
    for $S \subseteq C$, $|S| \leq k$, $S \neq T$ **do**
      if $\frac{\left|\left\{\boldsymbol{x} \in TR \middle| |f_T(\boldsymbol{x}) - f_S(\boldsymbol{x})| < \epsilon\right\}\right|}{|TR|} > 1 - \delta$ **then**
        $G_k = G_k \setminus \{f_T\}$
      end if
    end for
  end for
  if $G_k = \emptyset$ **then**
    $k = k - 1$
  else
    break
    return k
  end if
end while

Roughly speaking, for a small ϵ and δ close to 0, this means that, with only a few exceptions, the values of f_A and f_B are almost the same on the training data (we recommend $\epsilon = \delta = 0.05$ are reasonable default parameters). In order to find a proper upper bound k^*, the principle (6.9) can be used as before, just replacing coverage with ϵ-δ-coverage. The corresponding results are presented in Table 7.11. These results can be compared with the results in Table 7.3.

In order to find the proper k the Algorithm 1 is used in the experimental setup: Basically, the algorithm in each step constructs a set of features corresponding to k-additivity level (G_k). In the first step it starts with the features which were derived from biggest subset of C, namely in starting it considers only f_C. Then the algorithm searches in all features, which have cardinality smaller than $|C|$. If a feature that satisfies the ϵ, δ property is found, the algorithm removes f_C and goes one level deeper, namely to level $|C| - 1$. Then it checks, whether for all features f_T in $G_{|C|-1}$, there a feature f_S, exists with $|S| \leq |T|$ and $S \neq T$, so that can satisfy ϵ, δ property. Iteratively algorithm updates the set $G_{|C|-1}$ and in the end checks whether $G_{|C|-1} = \emptyset$. If it is the case, the algorithm goes one level deeper and performs the same procedure, otherwise it terminates and returns k.

6.3.2 2-additive Choquet Integral

In this section, we propose two approaches for reducing this complexity in the specific though practically relevant case of the 2-additive Choquet integral. Apart from theoretical results, we also present an experimental study in which we compare the two variants with the original implementation of choquistic regression. In the following, we restrict ourselves to the specific case of 2-additive fuzzy measures. This restriction is interesting for several reasons. In particular, one may of course hope for a gain in terms of computational efficiency. Besides, however, let us mention that a restriction of this kind is also interesting from a learning point of view: By allowing one to capture pairwise interactions between attributes, the 2-additive case is a proper generalization of the linear model, while at the same time, it is still reasonable in terms of the number of degrees of freedom. In fact, while the number of parameters to be estimated is exponential (in the number of attributes) in general, it is only quadratic in the 2-additive case. Practically, we could observe that the high flexibility of the general model is rarely needed; on the contrary, it often leads to problems of over-fitting the data, thereby compromising generalization performance.
Coming back to the computational aspect, the number of parameters to be estimated

is indeed reduced, since $m(A) = 0$ for all $A \subseteq C$ such that $|A| > 2$. On the other hand, it is important to observe that the number of constraints does not reduce: Although the number of summands in each of the constraints (4.12) becomes smaller (since many of them are now 0), the number of constraints themselves remains the same. In the following, we shall therefore look for ways to exploit the simplified structure of the 2-additive case in order to reduce the number of constraints.

Alternative Formulation I

Like before let $C = \{c_1, \ldots, c_m\}$ and let M denote the class of nonnegative monotone set functions on C, i.e., the class of functions $\nu : 2^C \to [0, +\infty)$ such that $\nu(A) \leq \nu(B)$ for all $A \subseteq B \subseteq C$; for the time being, we neglect the normalization condition, as it is less important for our purpose (it constitutes a single constraint that must be added to the optimization problem in order to turn a monotone measure into a fuzzy measure). More specifically, we are interested in the subclass $M_2 \subset M$ of 2-additive measures ν, i.e., whose Möbius transform satisfies $m_\nu(A) = 0$ for all $A \subseteq C$ such that $|A| > 2$.
The following characterization is well-known (see, e.g., Proposition 1 in [75]): $\nu \in M_2$ if and only if the following constraints $C_{i,X}$ are satisfied for all $c_i \in C$ and $X \subseteq C_i = C \setminus \{c_i\}$:

$$C_{i,X} : m_i + \sum_{c_j \in X} m_{i,j} \geq 0 \ , \tag{6.11}$$

where $m_i = m_\nu(\{c_i\})$ and $m_{i,j} = m_\nu(\{c_i, c_j\})$. Note that the number of constraints (6.11) is still exponential in m. Yet, we can show that they can be expressed equivalently in terms of a smaller number of constraints (albeit at the expense of introducing additional variables).

Proposition 6.2 *Condition (6.11) is equivalent to the following condition: For all $c_i \in C$, there exist $\alpha_{i,j} \in \mathbb{R},\ c_j \in C_i$, such that*

$$\begin{aligned} \alpha_{i,j} &\geq 0 \\ \sum_{\{j | c_j \in C_i\}} \alpha_{i,j} &\leq 1 \\ m_i &\geq 0 \\ m_{i,j} &\geq -\alpha_{i,j} \cdot m_i \end{aligned} \tag{6.12}$$

Proof 6.2 *Let $\nu \in M_2$ and suppose (6.11) to hold. For $c_i \in C$, (6.11) with $X = \emptyset$ implies $m_i \geq 0$. Now, define $C_i^- = \{j \mid c_j \in C_i, m_{i,j} < 0\}, C_i^+ = \{j \mid c_j \in C_i, m_{i,j} \geq 0\}$, and let*

$$f(n) = \begin{cases} 0, & \text{if } j \in C_i^+ \\ \frac{|m_{i,j}|}{m_i}, & \text{if } j \in C_i^- \end{cases} \tag{6.13}$$

Since (6.11) holds with $X = C_i^-$, we have

$$\sum_{j \in C_i^-} |m_{i,j}| \leq m_i \tag{6.14}$$

and therefore

$$\sum_{j \in C_i} \alpha_{i,j} = \sum_{j \in C_i^-} \alpha_{i,j} = \sum_{j \in C_i^-} \frac{|m_{i,j}|}{m_i} = \frac{1}{m_i} \sum_{j \in C_i^-} |m_{i,j}| \leq 1$$

Moreover, $m_{i,j} \geq -\alpha_{i,j} \cdot m_i$ holds by definition, both for $j \in C_i^+$ and $j \in C_i^-$. Thus, condition (6.12) holds, and hence (6.11) implies (6.12). Now, suppose that (6.12) holds. Then, $m_i \geq 0$ and for any $\emptyset \neq X \subseteq C_i$,

$$\begin{aligned} m_i + \sum_{\{j | c_j \in X\}} m_{i,j} &\geq m_i + \sum_{\{j | c_j \in X\}} -\alpha_{i,j} \cdot m_i \\ &= m_i - m_i \sum_{\{j | c_j \in X\}} \alpha_{i,j} \\ &= m_i (1 - \sum_{\{j | c_j \in X\}} \alpha_{i,j}) \geq 0 \end{aligned} \tag{6.15}$$

Thus, condition (6.11) holds, and hence (6.12) implies (6.11). ■

As a consequence of the above result, the constraints (6.11) can be replaced by the equivalent constraints (6.12). Thus, the number of constraints can indeed be reduced from exponential to quadratic, namely to $2m^2$ inequalities. On the other hand, (6.12) also comes with a disadvantage:While the constraints (6.11) are all linear, some of the constraints (6.12) are nonlinear (albeit convex); indeed, recall that the $\alpha_{i,j}$ are introduced as new variables that need to be determined simultaneously with the m_i and $m_{i,j}$.

Alternative Formulation II

Our second reformulation of the problem is based on a theoretical result showing that the class $\mathcal{M}_2$ or, more specifically, the class of normalized measures in $\mathcal{M}_2$

(i.e., those ν whose Möbius function additionally satisfies

$$\sum_{i=1}^{m} m_i + \sum_{1 \leq i < j \leq m} m_{i,j} = 1 \ , \tag{6.16}$$

forms a convex polytope. The extreme points of this polytope are exactly those $\{0,1\}$-valued measures whose Möbius transforms are of the form

$$m_A(X) = \begin{cases} 1, & \text{if } X = A \\ 0, & \text{otherwise} \end{cases} , A \in \mathcal{E} \tag{6.17}$$

or of the form

$$m_B^{'}(X) = \begin{cases} 1, & \text{if } \emptyset \neq X \subset B \\ -1, & \text{if} X = B \\ 0, & \text{otherwise} \end{cases} , A \in \mathcal{E}' \tag{6.18}$$

where $\mathcal{E} = \left\{ A \subseteq C \;\middle|\; 1 \leq |A| \leq 2 \right\}$ and $\mathcal{E}' = \left\{ B \subseteq C \middle|\; |B| = 2 \right\}$ [73]. In other words, each feasible solution m can be written as a convex combination of these m^2 extreme points:

$$m = \sum_{A \in \mathcal{E}} \alpha_A \cdot m_A + \sum_{B \in \mathcal{E}'} \alpha'_B \cdot m'_B \tag{6.19}$$

Consequently, the constraints (6.11), (6.16) can be replaced by (6.19) in conjunction with the following constraints:

$$\begin{aligned} &\alpha_A \geq 0 \\ &\alpha'_B \geq 0 \\ &\sum_{A \in \mathcal{E}} \alpha_A + \sum_{B \in \mathcal{E}'} \alpha'_B = 1 \end{aligned}$$

Like in our first reformulation, the number of constraints is thus significantly reduced, this time even without introducing nonlinearities, albeit again at the cost of a quadratic number of additional variables. More concretely, we end up with m^2 additional variables while reducing the number of constraints to $m^2 + 1$.

6.4 Measure Correction - From non Monotone Measure to Monotone Measure

Thus far the algorithms proposed in this thesis, consider monotonicity, i.e., the algorithms enforce monotonicity by auxiliary constraints. Since there are $m2^{m-1}$ constraints for assuring monotonicity (given m attributes), for large number of attributes solving the optimization problem indeed is quite difficult and time consuming. Chapter 1 clearly states that this thesis provides several algorithms and approaches to deal with monotone data. Since monotonicity in our case is a prerequisite, the datasets which are desirable are monotone. This means that, they have monotone structures. The ultimate goal of learning is to bias the observations. In this regard, the expectation is that the model can capture the properties of data. Since monotonicity is a kind of data-property, the model can capture such properties. Note that there is no guarantee, that the model can capture whole monotonicity property, but at least there is a chance to capture it partially. To this end, this section serves the idea of learning the monotone models underlying the Choquet integral without any monotonicity constraints. This problem in the literature is addressed under "relaxation". Basically the core idea of relaxation is to learn the optimal parameters for a fuzzy measure without enforcing any monotonicity constraints and finally making corrections for the optimal learned parameters. It is quite obvious, that there is no guarantee, that learned parameters satisfy monotonicity. Now the non trivial question is, how is possible to monotonize the learned parameters in the end? Let us consider more in the details the structure of fuzzy measure. Basically, the structure of a fuzzy measure can be seen as a DAG (directed acyclic graph) structure ($\mathfrak{S}$), where $V = \{V_1, \ldots, V_p\}$, the vertices, correspond to the elements of $\mathcal{P}(\{c_1, \ldots, c_m\})$, and moreover $E = \{(V_i, V_j) \mid V_i \subset V_j\}$ is the set of directed edges from V_i to V_j. We use the notation $V_i \preceq V_j$ if $(V_i, V_j) \in E$. Henceforth we assign to each vertex V_i the optimal learned parameter $\mu^*(V_i)$ (which does not necessarily obey monotonicity constraints) and the goal is to find the fuzzy measure $\mu^{**}(\cdot)$, where has a minimal distance to the original measure given a metric space. More precisely, the goal is as follows

$$\arg\min_{\mu^{**}} \left\{ d(\mu^{**}, \mu) \middle| \mu^{**} \text{ is a fuzzy measure on } C \right\} \ ,$$

where the distance function $d(\cdot, \cdot)$ is already given.
Generally three types of metrics are taken into account in the literature, namely, L_1, L_2 and L_∞. In general, the solution of a DAG structure problem is widely addressed in the literature. The problem is to find a set of values which are in ac-

cordance with the structure of a DAG (if $(V_i, V_j) \in E$ then $v(V_i) \leq v(V_j)$), and also has a minimal distance to the original values. Note that in general there is no unique solution to this problem.

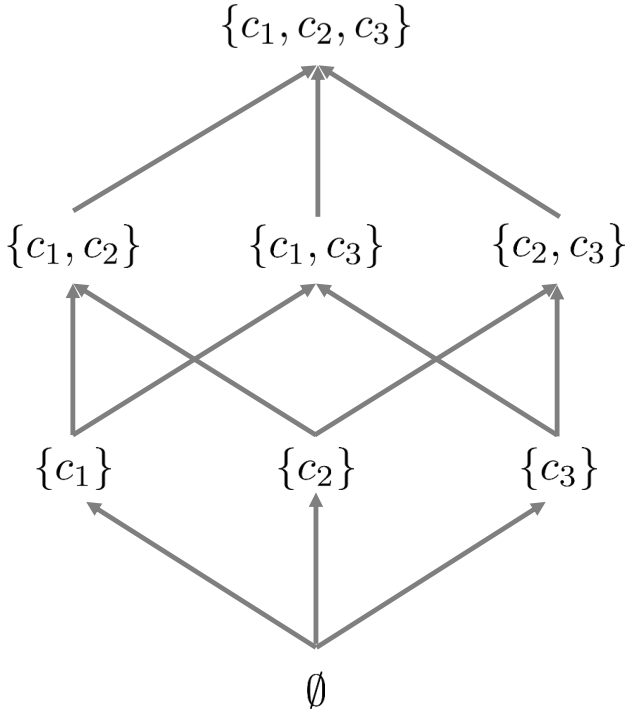

Figure 6.2: Directed acyclic graph structure representing the monotone relationship for three criteria. The directed edges show the direction of monotonicity, meaning the measure of the subset is smaller than the measure of the set.

In this regard, Jewell in [66] proposed the initial solution regarding the isotonic optimization. Assume we are given $\mathfrak{Y} = \{y_1, \ldots, y_p\}$ as p input values corresponding to the structure $\mathfrak{S}$. Moreover suppose $\mathfrak{f} = \{f_1, \ldots, f_p\}$ is the optimal solution in terms of distance. For every set $\mathfrak{f}$, which is in agreement with structure $\mathfrak{S}$, the different distances can be expressed as follows:

$$Err_z(\mathfrak{Y}, \mathfrak{f}) = \sum_{i=1}^{p} L_z(y_i - f_i) \quad z \in \{1, 2\}$$

and for L_∞

$$Err_z(\mathfrak{Y}, \mathfrak{f}) = \max_{i=1,\ldots,p} L_1(y_i - f_i) \quad z = \infty$$

Additionally he assumed a more general form for domination by presuming the following bounds for the optimal solutions:

$$\begin{aligned} y_j - y_i &\geq R_{ij} \\ y_j - y_i &\leq S_{ij} \\ A_i \leq y_i &\leq B_i \end{aligned}$$

Then the optimal solution $\mathfrak{f}^*$ for L_1 as well as the L_2 norm is equal to:

$$\begin{aligned} \mathfrak{f}^* &= \min_{\mathfrak{f}} Err_z(\mathfrak{Y}, \mathfrak{f}) \qquad\qquad z \in \{1, 2\} \\ &\text{s.t.} \\ R_{ij} + f_i + f_j \leq v_j - v_i &\leq S_{ij} + f_i - f_j \quad (v_i, v_j) \in E \\ A_i - f_i \leq v_i &\leq B_i - f_i \quad i \in \{1, \ldots, p\} \end{aligned}$$

Where $A = \big\{(i,j) \mid (v_i, v_j) \in E\big\}$.

Additionally for L_∞ the optimization problem is defined as follows:

$$\begin{aligned} \mathfrak{f}^* &= \min_{\mathfrak{f}} e \\ &s.t. \\ e - L_1(y_i, f_i) &\geq 0 \quad i \in \{1, \ldots, p\} \end{aligned}$$

This kind of optimization is called flow network optimization.
With respect to our setting assume the measure μ is given. Then the goal is to find μ^{**} as follows:

$$\begin{aligned} &\arg\min_{\mu^{**}} Err_z(\mu^{**}, \mu) && z \in \{1, 2, \infty\} \\ &\text{s.t.} \\ &\mu^{**}(V_j) - \mu^{**}(V_i) \geq R_{ij} && \forall (V_i, V_j) \in E \\ &\mu^{**}(V_j) - \mu^{**}(V_i) \leq S_{ij} && \forall (V_i, V_j) \in E \end{aligned}$$

Referring to the Subsection 3.3.3, there are $3^m - 2^m$ (m is the number of criteria) constraints in total. Note that, without loss of generality the R_{ij}, S_{ij} can be considered as $R_{ij} = 0$ and $S_{ij} = \infty$.
Stout in [93] proposed the idea for reduction the complexity of the flow framework above. The basic idea is to embed the DAG structure $\mathfrak{S}$ to a larger space, where there are fewer edges needed for representing the original structure $\mathfrak{S}$. He proved, by using this embedding that given p points in $d-$dimensional space ($d \geq 3$), the isotonic regression has a computational complexity of

- $\mathcal{O}(p^2 \log^d p)$ for L_1 metric
- $\mathcal{O}(p^3 \log^{2d-1} p)$ for L_2 metric
- $\mathcal{O}(p \log^d p)$ for L_∞ metric

Note that in our setting, $p = 2^m$, where m is the number of criteria.
In particular, the isotonic regression for unweighted data and L_∞ metric, is computed as follows [94]:

$$\mathfrak{f}_i^* = \frac{\max\left\{y(V_j) \mid V_j \preceq V_i\right\} + \min\left\{y(V_k) \mid V_i \preceq V_k\right\}}{2} \ ,$$

where $y(V_i)$ is the response value for the vertex V_i. Moreover, the above algorithm has complexity of $\mathcal{O}(e)$, where e is the number of edges. In our setting, given set $C = \{c_1 \ldots, c_m\}$, each $c_i \in C$ is connected to $\{c_i, T\}$, where $T \subseteq C \setminus \{c_i\}$. Therefore for each c_i, there exist $\sum_{k=1}^{m-1} \binom{m-1}{k}$ edges. In total there are $m \sum_{k=1}^{m-1} \binom{m-1}{k}$ edges, i.e., our setting has a computational complexity of $\mathcal{O}\big(m \sum_{k=1}^{m-1} \binom{m-1}{k}\big)$.

Burdakov et al. proposed in [19, 20] an algorithm based on a PAV (Pool-Adjacent-Violator) algorithm. They proposed the generalization of PAV called a GPAV for the purpose of multidimensional isotonic regressions. The basic idea is to collect the vertices by some clusters and update the cluster by adding the vertex, which violates the monotonicity constraints. They claim that the GPAV algorithm has computational complexity of $\mathcal{O}(p^2)$, where p is the number of vertices. In our setting, there are 2^m vertices in total.

In [41] an idea based on PAV for multidimensional cases has been proposed. Since every finite set of d-dimensional points can be represented by a union of DAG structures (by a Pareto dominance relation), in this case, the core idea is to decompose DAG structures into several maximal sub structures, in which each sub structure can be represented by the total order in terms of Pareto dominance. Then the PAV algorithm can be used for each sub structure, and after modification of the values, the next sub structure will be again modified by updated values. For our purposes since we are interested in all maximal the total orders, $m!$ total orders should be taken into consideration. Hence, the algorithm has a computational complexity of $\mathcal{O}(m!)$, where m is the number of criteria. For instance, the following order can be seen as a sub structure for a given $C = \{c_1, c_2, c_3\}$:

$$\{c_1\} \prec \{c_1, c_2\} \prec \{c_1, c_2, c_3\}$$

Block et al. also proposed an algorithm in [17] for a partially ordered isotonic regression under an L_2 norm.The basic idea is to change the partial order to a linear extension (which is not necessarily unique). Then they start with the first element and go successively through the sequence and in each step check whether for a given element V_i the magnitude of adjacent of V_i is in agreement with V_i. If it is not the case, the algorithm unifies V_i with its adjacent, makes it as one block and recomputes the corresponding value of this block. The algorithm terminates when there is no inconsistency in the sequence. Also in [83] Pardalos and Xue presented the **IRT-BIN** algorithm to tackle the isotonic regression problem for DAG structure. In this case, the core idea is the same as in [17]. Roughly speaking they start with a linear ordering of partial order, which is not necessarily unique. For each node V_i, they defined a block $B(V_i) = \{V_i\}$ and a binomial heap H_i. They start with the first element in this order and go through the sequence successively. For step i, the algorithm checks, whether $Av(B(x_i)) < Maximum(H_i)$. If it is the case, let $B(x_k)$ be the corresponding value to $\arg max_i(H_i)$. Then the algorithm unifies blocks $B(x_i)$ and $B(x_k)$ together as block $B(x_i)$ and assigns value $Av(B(x_i))$ to this block. Moreover, the algorithm merges heaps H_i and H_k to H_i heap. In each step, the algorithm updates the keys of blocks, heaps and $Av(\cdot)$. Finally the output is an isotonic regression. The time complexity of this algorithm is $\mathcal{O}(p \log p)$, where p is the number of nodes. Hence for our setting has computational complexity of $\mathcal{O}(m2^m \log 2)$.

The above approaches are addressed as solutions to the isotonic regression problem. However, from another point of view, the correction of the inconsistencies in the measure can be addressed in the literature by *minimal reassignment* under the

L_1 norm. Traditionally the minimal reassignment problem comes from the ordinal classification, in which some instances and their labels are given, and the task is to monotonize the data through minimal changing of labels. In essence, the task is to find new classes for given instances by minimum changing. In this regard, Dembczyński [30] and Feelder [35] proposed two approaches to tackle this problem. Before going into details, the following theorem should be introduced:

Theorem 6.4 *Assume μ is an arbitrary measure on set C. Moreover assume*

$$\Gamma = \left\{ \mu^{\bullet} \middle| \mu^{\bullet} \text{ is closest monotone measure to } \mu \text{ in terms of } L_1 \text{ metric} \right\}.$$

Then there exists $\mu^* \in \Gamma$, which has following property:

$$Im(\mu^*(.)) \subseteq Im(\mu(.))$$

This means that by rearranging the original values of response, the optimal solution can be found.

Proof 6.3 *(proof by negation) Assume there is no such measure. Let us assume $A \subset C$ is the smallest subset (minimal, note that the subset A is not unique) in terms of cardinality, such that*

$$\forall E \subseteq C \quad \mu^*(A) \neq \mu(E)$$

Therefore, we have

$$\max_{K \subset A} \mu^*(K) < \mu^*(A) \leq \min_{A \subset L} \mu^*(L)$$

The proof can be followed in following steps:

- *If*

  $$\mu(A) \leq \max_{K \subset A} \mu^*(K)$$

  then by defining

  $$\mu^{\diamond}(S) := \begin{cases} \mu^*(S) & S \neq A \\ \max_{K \subset A} \mu^*(K) & S = A \end{cases}$$

  the updated measure $\mu^{\diamond}$ is closer to the μ, which apparently is a contradiction. The reason is $|\mu^{\diamond}(A) - \mu(A)| < |\mu^*(A) - \mu(A)|$. Note that in this case $\mu(A) \leq \max_{K \subset A} \mu^*(K) < \mu^*(A)$.

- *If*

$$\max_{K\subset A}\mu^*(K) < \mu(A) < \min_{A\subset L}\mu^*(L)$$

then by defining

$$\mu^\diamond(S) := \begin{cases}\mu^*(S) & S \neq A\\ \mu(A) & S = A\end{cases}$$

the updated measure $\mu^\diamond$ is closer to the μ, which surely is in contradiction to the assumption.

- *If*

$$\max_{K\subset A}\mu^*(K) < \mu^*(A) < \min_{A\subset L}\mu^*(L) < \mu(A)$$

then by defining

$$\mu^\diamond(S) := \begin{cases}\mu^*(S) & S \neq A\\ \min_{S\subset L}\mu^*(L) & S = A\end{cases}$$

the measure $\mu^\diamond$ is closer to the μ. Hence there is apparently a contradiction again.

- *If*

$$\max_{K\subset A}\mu^*(K) < \mu^*(A) = \min_{A\subset L}\mu^*(L) < \mu(A)$$

then there exists $\tilde{A}$, $A \subset \tilde{A}$ $s.t.$ $\mu^*(A) = \mu^*(\tilde{A})$. Assume A_U is the largest (maximal) subset in terms of cardinality (note that the subset A_U in general is not unique), which has the above property. In other words:

$$\forall F \ \ s.t. \ \ A_U \subset F \ \ \Rightarrow \mu^*(A_U) < \mu^*(F)$$

Moreover let $\mathfrak{T} = \{T \subseteq C \mid A \subseteq T \subseteq A_U\}$.

It is easy to check that, it is not possible that $\forall B \in \mathfrak{T}$

$$\mu(B) < \mu^*(B)$$

It is also easy to check that, it is not possible that $\forall B \in \mathfrak{T}$

$$\mu^*(B) < \mu(B)$$

(Otherwise, by defining $\mu^*(A) := \max_{B\in\mathfrak{T}} \mu(B)$ *and* $\mu^*(A) := \min_{B\in\mathfrak{T}} \mu(B)$ *respectively, is in contradiction to our assumption.) Additionally*

$$\sum_{B\in\mathfrak{T}} |\mu(B) - \mu^*(B)| = \sum_{B\in\mathfrak{T}} |\mu(B) - C|$$

Suppose $B^* = \arg min_{B\in\mathfrak{T}} |\mu(B) - C|$*. By defining* $\mu^\diamond(.)$ *as follows:*

$$\mu^\diamond(S) := \begin{cases} \mu^*(S) & S \in C \setminus \mathfrak{T} \\ \mu(B^*) & S \in \mathfrak{T} \end{cases}$$

the measure $\mu^\diamond(.)$ *contains only the values from the original measure, and has the minimal distance to the original measure, which means it is in agreement with measure* μ^**.*

Note that in last step there is no contradiction. The core idea, is to shift a subset of optimal monotone measure, by preserving the distance. In other words, the new measure is redefined, in a way that has same distance to original measure μ*, where the redefined values belong to original measure* μ*.* ■

The above theorem shows, from an application point of view, that the values of an optimal monotone measures can be chosen among the values of the original measure. The aforementioned property allows us to use methods like minimal reassignment labeling proposed in [30, 35]. More precisely, the core idea is to rearrange the values in the original measure so that the new measure is monotone and also has a minimal distance to the original measure under the L_1 norm. In the following we use an example to show how the optimal solution looks under L_1 for 2 different approaches:

$\mu(\{c_1\}) = .2$	$\mu^*(\{c_1\}) = .2$	$\mu^{**}(\{c_1\}) = .2$
$\mu(\{c_2\}) = .5$	$\mu^*(\{c_2\}) = .4$	$\mu^{**}(\{c_2\}) = .45$
$\mu(\{c_3\}) = .5$	$\mu^*(\{c_3\}) = .4$	$\mu^{**}(\{c_3\}) = .45$
$\mu(\{c_1, c_2\}) = .4$	$\mu^*(\{c_1, c_2\}) = .4$	$\mu^{**}(\{c_1, c_2\}) = .45$
$\mu(\{c_1, c_3\}) = .4$	$\mu^*(\{c_1, c_3\}) = .4$	$\mu^{**}(\{c_1, c_3\}) = .45$
$\mu(\{c_2, c_3\}) = .6$	$\mu^*(\{c_2, c_3\}) = .6$	$\mu^{**}(\{c_2, c_3\}) = .6$
$\mu(\{c_1, c_2, c_3\}) = 1$	$\mu^*(\{c_1, c_2, c_3\}) = 1$	$\mu^{**}(\{c_1, c_2, c_3\}) = 1$

Here the μ is referred to as a non monotone measure, μ^* is the optimal solution by minimal reassignment labeling approach and μ^{**} is referred to as the optimal solution by the PAV algorithm. As it can be seen $d(\mu, \mu^*) = d(\mu, \mu^{**}) = .2$.
Now assume a measure μ is given which does not obey monotonicity properties. In order to use minimal reassignment labeling, the basic idea is to order values, and label them by $\{1, \ldots, L\}$. In the next step, the following transformation is used:

$$t : \mathcal{P}(\{c_1 \ldots, c_m\}) \to \{0,1\}^m$$
$$t(A) = \big(t_1(A), \ldots, t_m(A)\big)$$

Where

$$t_i(A) = \begin{cases} 1 & \text{if } c_i \in A \\ 0 & \text{otherwise} \end{cases}$$

It is easy to show that the mapping $t(\cdot)$ is well-defined. Note that the transformation holds the information of monotonicity dependencies, which in essence is necessary. After transforming the measure and values, one can deal with the following setting:

$$\Big\{(t(A), L_A)\Big\}_{A \in C} \subset \{0,1\}^m \times \{1, \ldots, L\}$$

This is indeed the structure, that is used in minimal reassignment labeling. Suppose the optimal solution is computed by minimal reassignment labeling as follows:

$$\Big\{(t(A), L_A^*)\Big\}_{A \in C} \subset \{0,1\}^m \times \{1, \ldots, L\}$$

Then it is needed to use the inverse function $t^{-1}(\cdot)$ and inverse ordering function $o^{-1}(\cdot)$ to get the fuzzy measure as follows:

$$\Big\{(t^{-1}(t(A)), o^{-1}(L_A^*))\Big\}_{A \in C} = \Big\{(A, \mu^*(A))\Big\}_{A \in C}$$

Heuristic Approach

Thus far all mentioned approaches present the optimal solutions under predefined metrics. However, one can also imagine, when the majority of measure are fulfilled monotonicity constraints, some heuristic methods can also solve the problem in a meaningful manner. In the following, we introduce a method, to tackle this problem in an approximate manner. Before going into details we should introduce some preliminaries.

Let μ be an arbitrary measure. Then $\mu_{min^e_{clo}}$ and $\mu_{max^e_{clo}}$, the strict $\min$ and $\max$ closure for measure μ are defined as follows:

$$\mu_{\min^e_{clo}}(A) := \min\Big\{\mu(B) \mid A \subseteq B\Big\}$$

$$\mu_{\max^e_{clo}}(A) := \max\Big\{\mu(B) \mid B \subseteq A\Big\}$$

As can be seen, $\mu_{\min^e_{clo}}$ and $\mu_{\max^e_{clo}}$ are monotone. The following theorem shows that $\min$ and $\max$ closure are bounds for the optimal monotone measure.

Theorem 6.5 *Let μ be an arbitrary measure on set C (not necessarily monotone measure) and moreover let*

$$\mu^* \in \Big\{d(\mu, \mu^{'})|\mu^{'} \textit{ is a monotone measure on } C\Big\} \ , \tag{6.20}$$

where $d(\cdot,\cdot)$ is a distance function. Then $\forall\ A\ \subseteq C$

$$\mu_{min^e_{clo}}(A) \leq \mu^*(A) \leq \mu_{max^e_{clo}}(A) \ , \tag{6.21}$$

where $\mu_{\min^e_{clo}}$ and $\mu_{\max^e_{clo}}$ are lower and upper closures with respect to measure μ.

Proof 6.4 *(Proof by negation) Before going into the details, first of all note that measures $\mu_{\min^e_{clo}}$ and $\mu_{\max^e_{clo}}$ are monotone. Also for every subset A of C, the following inequality is valid:*

$$\mu_{min^e_{clo}}(A) \leq \mu_{orig}(A) \leq \mu_{max^e_{clo}}(A). \tag{6.22}$$

Suppose $\mu^* \in \Big\{d(\mu, \mu^{'})|\mu^{'}$ is a monotone measure on $C\Big\}$ and the assumption in 6.21 is not valid for measure μ^*. Let us define measure $\mu^\bullet$ as follows:

$$\mu^\bullet = \begin{cases} \mu_{\min^e_{clo}}(A), & \textit{if } \mu^*(A) < \mu_{\min^e_{clo}}(A) \\ \mu_{\max^e_{clo}}(A), & \textit{if } \mu^*(A) > \mu_{\max^e_{clo}}(A) \\ \mu^*(A) & \textit{otherwise} \end{cases} \tag{6.23}$$

In the first case, we say that $\mu^*(A)$ is up-corrected, in the second case, that $\mu^*(A)$ is down-corrected. Obviously, $\mu_{\min_{clo}^e} \leq \mu^\bullet \leq \mu_{\max_{clo}^e}$.

We claim that $\mu^\bullet$ thus defined is a monotone measure. To see this, consider a path from the empty set to C in the form of a chain

$$\emptyset \subset A_0 \subset A_1 \subset \ldots \subset A_m = C \ ,$$

where $|A_k| = k$. Suppose $\mu^\bullet$ is not monotone along the chain.Then there is a k such that $\mu^\bullet(A_k) > \mu^\bullet(A_{k+1})$. Since μ^* is monotone, $\mu^*(A_k) \leq \mu^*(A_{k+1})$. Therefore, either $\mu^*(A_k)$ has been up-corrected or $\mu^*(A_{k+1})$ has been down-corrected. In the first case,

$$\mu^\bullet(A_k) = \mu_{\min_{clo}^e}(A_k) \leq \mu_{\min_{clo}^e}(A_{k+1}) \leq \mu^\bullet(A_{k+1})$$

which is a contradiction. In the second case,

$$\mu^\bullet(A_{k+1}) = \mu_{\max_{clo}^e}(A_{k+1}) \geq \mu_{\max_{clo}^e}(A_k) \geq \mu^\bullet(A_k)$$

which is again a contradiction. Therefore, $\mu^\bullet$ is monotone.
Now, suppose that (6.22) does not hold. Then, there will be at least one up- or down-correction according to (6.23), and since each correction moves $\mu^\bullet(A_i)$ closer to $\mu(A_i)$ (as compared to $\mu^*(A_i)$), it follows that $d(\mu, \mu^\bullet) < d(\mu, \mu^*)$. In conjunction with the monotonicity of $\mu^\bullet$, this contradicts (6.20). ■

So as mentioned earlier, satisfying monotonicity constraints in proposed method has two phases. First the majority of constraints will be satisfied through the structure of data. Then in next step, we make a correction to have a monotone measure. To this end, given a measure we propose its approximation, namely

- lower closure by $\min$ operator
- upper closure by $\max$ operator

A non-trivial question however is, which correction should be taken into account? To this end, we take a convex combination of these two corrections and try to minimize the distance between the convex combination and the original measure;

$$P^* = \min_{0 \leq P \leq 1} D\Big(\boldsymbol{\mu}_{orig}, P\boldsymbol{\mu}_{min^e_{clo}} + (1-P)\boldsymbol{\mu}_{max^e_{clo}}\Big) \tag{6.24}$$

The last constraint can be omitted, since can be adjusted manually. This means if the output is larger than 1 or less than 0, we put it to 1 and 0 respectively. Obviously this convex combination is also a monotone measure and by a value of P we can minimize the distance to the original measure. After correcting the measure, the obtained fuzzy measure can be used by the support vector machine in primal form. This optimization problem has computational complexity of $\mathcal{O}(p)$ under L_2 norm, where p is the number of vertices (in this case 2^m). The basic reason is, computing $\mu_{min_{clo}}$ and $\mu_{max_{clo}}$ is linear in the number of vertices and also for the optimization step, since there is only one parameter, the optimization needs linear time complexity. More precisely, the optimal P for L_2 norm can be determined as follows:

$$\frac{\sum_{A \subseteq C} \Big\{\mu_{max^e_{clo}}(A) + \mu_{orig}(A) - \mu_{orig}(A)\mu_{max^e_{clo}}(A) - \mu_{max^e_{clo}}(A) \cdot \mu_{min^e_{clo}}(A)\Big\}}{\sum_{A \subseteq C} \Big(\mu_{min^e_{clo}}(A) - \mu_{max^e_{clo}}(A)\Big)^2}$$

In cases, when $P > 1$ or $P < 0$, values 1 and 0 are taken respectively. In our experiments we take the L_2 norm as distance D. The corresponding results for the heuristic method as well IRT-BIN algorithm are presented in Table 7.8. Additionally the optimization setting in (6.24), is completely in agreement with L_∞ norm, where the optimal value P is equal to 0.5; however as mentioned the optimal solution is not unique.

The Usefulness of Measure Correction

As mentioned the core idea is to learn the parameters without any monotonicity constraints. To this end, both setting, with respect to maximum likelihood and kernel-based learning approaches are taken into consideration. For maximum likelihood setting the following objective function and constraints are considered:

$$\max_{\boldsymbol{m},\gamma,\beta} \left\{ -(1-y)\,\gamma \sum_{i=1}^{n} (\mathcal{C}_{\boldsymbol{m}}(\boldsymbol{x}_i) - \beta) - \sum_{i=1}^{n} \log\Big[1 + \exp\left(-\gamma\,\mathcal{C}_{\boldsymbol{m}}(\boldsymbol{x}_i) - \beta\right)\Big] \right\} \tag{6.25}$$

s.t.

$0 \leq \beta$

$0 < \gamma$

in which the $C_{\boldsymbol{m}}(\cdot)$, β and γ are refer to the notations in Section 4.1.2.
For the kernel-based approach the following setting is taken into account:

$$\min_{\boldsymbol{\alpha}} \left\{ \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K_{\mathcal{C}}^{k=p}(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i=1}^{n} \alpha_i \right\} \tag{6.26}$$

s.t.

$$\sum_{i=1}^{n} y_i \alpha_i = 0$$

$$0 \leq \alpha_i \leq C \qquad \forall i \in \{1, \ldots, n\} \ ,$$

where $K_{\mathcal{C}}^{k=p}(\cdot, \cdot)$ is assigned to the Choquet kernel of degree p. More details about the setting can be found in Subsection 2.4.3. Note that although for the choquistic regression setting there is a possibility of ensuring monotonicity, for the Choquet kernel with dual from setting there is no way to assure monotonicity. Therefore, such a correction is quite desirable. The corresponding results are shown in Chapter 7, Table 7.8.

7 Data Sets and Experimental Parts

In this chapter, the data sets, which have been used during the experiments are introduced and the results related to each approach are presented. In the following, we describe the datasets that were primarily, from UCI repository [1] and WEKA [56]. The collection of data for experimental evaluation is a bit hindered by the fact that our models are monotone models. Data sets for which monotonicity of this kind is a reasonable assumption are less frequent than standard classification data. Parts of the results in this chapter were already published in [60, 97, 99, 98, 101].

7.1 Data Description

- **Employee Selection (ESL):** This data set contains profiles of applicants for certain industrial jobs. The values of the four input attributes were determined by expert psychologists based upon psychometric test results and interviews with the candidates. The output is an overall score on an ordinal scale between 1 and 9, corresponding to the degree of suitability of each candidate to this type of job. For binary classification purpose, we binarized the output value by distinguishing between suitable (score $6-9$) and unsuitable (score $1-5$) candidates, whereas for ordinal classification purpose the original labels were

[1] `http://archive.ics.uci.edu/ml/`

taken into account.

- **Employee Rejection/Acceptance (ERA):** This data set originates from an academic decision-making experiment. The input attributes are features of a candidate such as past experience, verbal skills, etc., and the output is the subjective judgment of a decision-maker, measured on an ordinal scale from 1 to 9, to which degree he or she tends to accept the applicant for the job. For binary classification purpose, we binarized the output value by distinguishing between acceptance (score $5-9$) and rejection (score $1-4$).

- **Lecturers Evaluation (LEV):** This data set contains examples of anonymous lecturer evaluations, taken at the end of MBA courses. Students were asked to score their lecturers according to four attributes such as oral skills and contribution to their professional/general knowledge. The output was a total evaluation of each lecturer's performance, measured on an ordinal scale from 0 to 4. In the case of binary classification, we binarized the output value by distinguishing between good (score $3-4$) and bad evaluation (score $0-2$).

- **Mammographic (MMG):** This data set is taken from breast cancer screening by mammography. The goal is to predict the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes (mass shape, mass margin, density) and the patient's age.

- **CPU:** This is a standard benchmark data set from the UCI repository. It contains eight input attributes, two of which were removed since they are obviously of no predictive value (vendor name, model name). The problem is to predict the (estimated) relative performance of a CPU (binarized by thresholding at the median) based on its machine cycle time in nanoseconds, minimum main memory in kilobytes, maximum main memory in kilobytes, cache memory in kilobytes, minimum channels in units and maximum channels in units.

- **Car Evaluation (CEV):** This data set contains 6 attributes describing a car, namely, buying price, price of maintenance, number of doors, capacity in terms of persons to carry, the size of the luggage boot and estimated safety of the car. The output is the overall evaluation of the car: unacceptable, acceptable, good, very good. For binary classification purpose, we binarized this evaluation into unacceptable versus acceptable and (very) good.

- **Breast Cancer (BCC):** This dataset was obtained from the University Medical Center, Institute of Oncology in Ljubljana, Yugoslavia. There are 7

attributes, namely, menopause gain, tumor-size, inv-nodes, node-caps, deg-malig, breast cost and irradiat gain. The output is a binary variable, namely, no-recurrence-events and recurrence-events.

- **DenBosch (DBS):** This data set contains 8 attributes describing houses in the city of Den Bosch: district, area, number of bedrooms, type of house, volume, storeys, type of garden, garage, and price. The output is a binary variable indicating whether the price of the house is low or high (depending on whether or not it exceeds a threshold).

- **Auto MPG:** This data set was used in the 1983 American Statistical Association Exposition. The attributes are: mpg, cylinders, displacement, horsepower, weight, acceleration, model year and origin. The data concerns city-cycle fuel consumption in miles per gallon, to be predicted in terms of 3 multivalued discrete and 5 continuous attributes. We removed incomplete instances and additionally for binary classification purpose we binarized the output by applying the median.

- **Social Workers Decisions (SWD):** The data set is about risk assessment of social workers facing the children at home, when their families stayed at home. It contains 10 ordinal input attributes and one ordinal output. For binary classification purpose we binarized the output by grouping labels $2, 3$ as negative class and $4, 5$ as a positive class.

- **Color yield:** Finally, we took data from an industrial polyester dyeing process that was also analyzed in [80]. Here, the output variable is the color yield, which has been measured as a function of three important factors: disperse dyes concentration, temperature and time of dyeing. Corresponding experiments have been made for seven different colors, giving rise to seven data sets. Each of these data sets was binarized by thresholding the color yield at its median value.

data set	#instances	#attributes	source
Yield Color 1-7	120	3	[80]
Employee Selection (ESL)	488	4	WEKA
Employee Rejection \ Acceptance (ERA)	1000	4	WEKA
Lecturers Evaluation (LEV)	1000	4	WEKA
Mamographic (MMG)	830	5	UCI
CPU	209	6	UCI
Car Evaluation (CEV)	1728	6	UCI
Breat-Cancer(BCC)	286	7	UCI
DenBosch	120	8	[26]
Auto MPG	398	8	UCI
SWD	1000	10	[9]

Table 7.1: Data sets and their properties

7.2 Normalization

Since the Choquet integral originally is constructed based on a combination of *copulas*, here namely $\min$ and the copula's domain is [0,1], it is necessary to normalize the input values. Note that, big difference in terms of magnitude (scale) for different features can effect on results (bias). First of all the normalization is meant to turn each predictor variable into a criterion, i.e., a "the higher the better" attribute, and to assure commensurability between the criteria [76]. As mentioned earlier, the criteria should range between 0 and 1. To this end, in the following we propound two types of normalization:

• The first normalization method is based on linear transformation, namely, given by the mapping $z_i = f_i(x_i) = (x_i - m_i)/(M_i - m_i)$, where m_i and M_i are lower and upper bounds for x_i (perhaps estimated from the data); if the influence of x_i is actually negative (i.e., $w_i < 0$), then the mapping $z_i = f_i(x_i) = (M_i - x_i)/(M_i - m_i)$ is used instead.
• The second normalization method is based on a cumulative distribution function, which is more robust to outliers and produces a more uniform distribution of normalized values. We therefore propose the mapping

$$z_i = F^{-1}(x_i) \ , \tag{7.1}$$

where F is the cumulative distribution function $x \mapsto \mathbf{P}(X_i \leq x)$. Of course, since this function is in general not known, it has to be replaced by an estimate $\hat{F}$; to this end, we simply adopt the empirical distribution of the training data (i.e., $\hat{F}(x)$ is the relative frequency of instances $\boldsymbol{x} = (x_1, \ldots, x_m)$ in the training data for which $x_i \leq x$).

7.3 Methods

All methods (and their abbreviations) which are used for the experiments are presented in Table 7.2.

Abbreviation	Method
CR	Choquistic Regression
LR	Logisitc Regression
KLR-Ply	Kernel Logistic Regression d=2
KLR-rbf	Kernel Logistic Regression RBF
MORE	MOnotone Rule Ensembles
LMT	Logistic Monotone Tree
CR-AI	Choquistic Regression (Nonlinear Monotonicity Constraints)
CR-AII	Choquistic Regression (Convex Combination Representation)
CK + CC	Choquet Kernel with Convex Combination (Measure Correction)
CK + IRT-BIN	Choquet Kernel with IRT-BIN (Measure Correction)
CR*	Choquistic Regression without Monotonicity Constraints
CR* + CC	Choquistic Regression with Convex Combination (Measure Correction)
CR* + IRT-BIN	Choquistic Regression with IRT-BIN (Measure Correction)
PLY d=a	Polynomial Kernel with Degree a
CK k=b	b-additive Choquet Kernel
CK K=n + CC	Choquet Kernel with Convex Combination (Measure Correction)
RBF	rbf kernel
OLR	Ordinal Logistic Regression
OCR	Ordinal Choquistic Regression
OCR + R	Ordinal Choquistic Regression with Hierarchical Regularization

Table 7.2: The methods which are used in this thesis for the experiments

7.4 Experimental Results Regarding Binary Class Classification

In this section, the results with respect to binary classification as well as setting are demonstrated.

Binary Classification by Choquistic Regression (Evaluation & Results)

Several experiments regarding the binary classification problem have been conducted using different settings and normalization methods in [98, 97, 99]. In this section only one of them is chosen thus allowing for comparisons.

- The experiment has been done by using a normalization method based on the cumulative distribution function. We assumed the following methods for comparison:

  Since choquistic regression (CR) can be seen as an extension of standard logistic regression (LR), it is natural to compare these two methods. Essentially, this comparison should give an idea of the usefulness of increased flexibility. On the other side, one may also ask for the usefulness of assuring monotonicity. Therefore, we additionally included two other extensions of LR, which are flexible but not necessarily monotone, namely kernel logistic regression (KLR) [114] with polynomial and Gaussian kernels. The degree of the polynomial kernel was set to 2, so that it models low-level interactions of the features. The Gaussian kernel, on the other hand, is able to capture interactions of a higher order. For each data set, the width parameter of the Gaussian kernel was selected from $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ in the most favorable way. Likewise, the regularization parameter η in choquistic regression was selected from
  $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$. Finally, we also included two methods that are both monotone and flexible, namely the MORE algorithm for learning rule ensembles under monotonicity constraints [29] and the LMT algorithm for logistic model tree induction [69]. Following the idea of forward stagewise additive modeling [103], the MORE algorithm treats a single rule as a subsidiary base classifier in the ensemble. The rules are added to the ensemble one by one. Each rule is fitted by concentrating on the examples that are most difficult to classify correctly by rules that have already been generated. The LMT algorithm builds tree-structured models that contain logistic regression functions at the leaves. It is based on a stagewise fitting process to construct the logistic regression models that can select relevant attributes from the data. This process is used to build the logistic regression models at the leaves by incrementally refining those constructed at higher levels in the tree structure.
  As performance measures, we determined the standard misclassification rate (0/1 loss). Estimates of this measure were obtained by randomly splitting the data into two parts, one part for training and one part for testing. This procedure was repeated 100 times, and the results were averaged. In order to analyze the influence of the amount of training data, we varied the proportion between training and test data from $20 : 80$ over $50 : 50$ to $80 : 20$. In these experiments, we used a variant of CR in which the underlying fuzzy measure

is restricted as to be k-additive, with k determined by means of an internal cross validation. Compared with other variants, this one performed best in terms of accuracy. A possible improvement of CR over its competitors, in terms of predictive accuracy, may be due to two reasons: First, in comparison to standard LR, it is more flexible and has the ability to capture nonlinear dependencies between input attributes. Second, in comparison to non-monotone learners, it takes background knowledge about the dependency between input and output variables into consideration. An overview of the results of the experiments is given in Table 7.3.

dataset	CR	LR	KLR-ply	KLR-rbf	MORE	LMT
ESL	.0682±.0129(1)	.0733±.0107(2)	.1488±.0278(6)	.0756±.0167(3)	.0838±.0241(5)	.0771±.0148(4)
ERA	.2889±.0273(1)	.2902±.0317(2)	.3001±.0130(5)	.2934±.0112(3)	.3155±.0150(6)	.2963±.0126(4)
LEV	.1499±.0122(1)	.1655±.0082(3)	.1627±.0119(2)	.1691±.0125(5)	.1707±.0186(6)	.1672±.0140(4)
MMG	.1725±.0120(1)	.1729±.0122(2)	.1960±.0160(6)	.1791±.0133(4)	.1764±.0137(3)	.1803±.0171(5)
CPU	.0811±.0103(3)	.0711±.0312(1)	.0996±.0231(6)	.0802±.0292(2)	.0829±.0379(4)	.0850±.0256(5)
CEV	.0448±.0089(3)	.1410±.0079(6)	.0663±.0130(5)	.0618±.0151(4)	.0339±.0076(1)	.0432±.0116(2)
BCC	.2775±.0335(2)	.2893±.0240(6)	.2760±.0243(1)	.2787±.0237(3)	.2827±.0255(4)	.2884±.0306(5)
DBS	.1713±.0424(2)	.2124±.0650(6)	.1695±.0437(1)	.1883±.0536(4)	.1932±.0511(5)	.1779±.0420(3)
MPG	.0709±.0193(1)	.0832±.0151(6)	.0788±.0097(4)	.0772±.0107(2)	.0811±.0119(5)	.0773±.0148(3)
avg. rank	1.67	**3.78**	**4**	3.33	**4.33**	**3.89**
ESL	.0601±.0126(1)	.0704±.0113(4)	.1023±.0225(6)	.0682±.0121(2)	.0695±.0139(3)	.0709±.0135(5)
ERA	.2844±.0306(1)	.2851±.0303(2)	.2926±.0151(4)	.2882±.0142(3)	.3037±.0180(6)	.2956±.0148(5)
LEV	.1372±.0125(1)	.1651±.0133(6)	.1520±.0160(4)	.1493±.0165(3)	.1486±.0157(2)	.1545±.0142(5)
MMG	.1667±.0144(1)	.1701±.0158(5)	.1721±.0164(6)	.1693±.0130(4)	.1691±.0140(3)	.1671±.0167(2)
CPU	.0464±.0281(1)	.0626±.0247(4)	.0835±.0264(6)	.0547±.0233(3)	.0489±.0226(2)	.0674±.0243(5)
CEV	.0376±.0059(4)	.1360±.0101(6)	.0328±.0057(3)	.0463±.0086(5)	.0215±.0053(2)	.0174±.0069(1)
BCC	.2687±.0282(4)	.2799±.0245(6)	.2591±.0287(1)	.2599±.0301(2)	.2640±.0288(3)	.2717±.0295(5)
DBS	.1572±.0416(4)	.1708±.0380(6)	.1333±.0333(1)	.1692±.0382(5)	.1457±.0413(3)	.1473±.0406(2)
MPG	.0577±.0251(1)	.0654±.0150(2)	.0728±.0159(4)	.0744±.0151(5)	.0751±.0178(6)	.0672±.0164(3)
avg. rank	2	**4.56**	**3.89**	3.56	3.33	3.67
ESL	.0542±.0218(1)	.0660±.0203(3)	.0922±.0279(6)	.0657±.0229(2)	.0661±.0219(4)	.0691±.0228(5)
ERA	.2813±.0280(1)	.2843±.0302(2)	.2918±.0290(5)	.2905±.0312(3)	.2988±.0276(6)	.2910±.0290(4)
LEV	.1314±.0176(1)	.1627±.0249(6)	.1472±.0231(3)	.1496±.0233(5)	.1397±.0214(2)	.1474±.0232(4)
MMG	.1584±.0251(1)	.1657±.0232(4)	.1741±.0246(6)	.1696±.0271(5)	.1645±.0235(3)	.1595±.0283(2)
CPU	.0212±.0301(1)	.0640±.0335(5)	.0754±.0372(6)	.0405±.0284(3)	.0412±.0299(4)	.0338±.0352(2)
CEV	.0273±.0089(4)	.1328±.0173(6)	.0286±.0075(5)	.0239±.0066(3)	.0190±.0070(2)	.0089±.0047(1)
BCC	.2496±.0485(1)	.2773±.0548(6)	.2569±.0506(2)	.2598±.0529(4)	.2570±.0463(3)	.2707±.0554(5)
DBS	.1416±.0681(4)	.1616±.0743(6)	.1265±.0663(2)	.1343±.0672(3)	.1242±.0609(1)	.1433±.0667(5)
MPG	.0551±.0160(1)	.0611±.0263(2)	.0727±.0268(4)	.0740±.0284(6)	.0737±.0269(5)	.0614±.0251(3)
avg. rank	1.67	**4.44**	**4.33**	**3.78**	3.33	3.44

Table 7.3: Classification performance in terms of the mean and standard deviation of 0/1 loss. From top to bottom: 20%, 50%, and 80% training data. (Average ranks comparing significantly worse with CR at the 90% confidence level are put in bold font.)

Moreover, a summary in terms of pairwise win statistics is provided in Table 7.4. As can be seen, CR compares quite favorably with the other approaches, especially with the non-monotone KLR methods, both in terms of 0/1 loss. It also outperforms LR, at least for sufficiently extensive training data; if the amount of training data is small, however, LR is even better, probably because CR will then tend to overfit the data. This is indeed a general trend that can be observed both for performance in terms of average ranks and the number of wins in pairwise comparison with another method: The more train-

	CR	LR	KLR-ply	KLR-rbf	MORE	LMT
CR	–	8 | 9 | 9	7 | 6 | 8	8 | 8 | 7	8 | 6 | 7	8 | 7 | 8
LR	1 | 0 | 0	–	4 | 5 | 5	5 | 2 | 3	5 | 2 | 3	5 | 4 | 3
KLR-ply	2 | 3 | 1	5 | 4 | 4	–	3 | 4 | 4	5 | 4 | 3	3 | 4 | 3
KLR-rbf	1 | 1 | 2	4 | 7 | 6	6 | 5 | 5	–	7 | 4 | 3	6 | 5 | 4
MORE	1 | 3 | 2	4 | 7 | 6	4 | 5 | 6	2 | 5 | 6	–	4 | 4 | 4
LMT	1 | 2 | 1	4 | 5 | 6	6 | 5 | 6	3 | 4 | 5	5 | 5 | 5	–

Table 7.4: Win statistics (number of data sets on which the first method was better than the second one) for 20%, 50%, and 80% training data for 0/1 loss case.

ing data is available, the better CR becomes, arguably because its flexibility is then becoming more and more advantageous. Needless to say, statistical significance is difficult to achieve due to the limited number of data sets. In terms of pairwise comparisons, for example, a standard sign test will not report a significant difference (at the 10 % significance level) unless one of the methods wins at least 7 of the 9 data sets. For the $0/1$ loss, this is indeed accomplished by CR in all cases except two (comparison with KLR-ply and MORE on 50 % training data); see Table 7.4. We also applied the two-step procedure recommended by Demsar [31], consisting of a Friedman test and (provided this one rejects the null-hypothesis of overall equal performance of all methods) the subsequent use of a Nemenyi test in order to compare methods in a pairwise manner; both tests are based on average ranks. For $0/1$ loss, the Friedman test finds significant differences among the six classifiers (at the 10 % significance level) when all three different proportions of data are used for training. The critical difference of ranks in the Nemenyi test is 2.28 for the $0/1$ measure. In Table 7.3, the average ranks for which this difference is exceeded are highlighted in bold font.

Binary Classification by Choquistic Regression (2-additive case)

- Experimentally, we compared three versions of the 2-additive choquistic regression the original formulation (CR-orig), the first reformulation (CR-AI), and the second reformulation (CR-AII) from Subsection 6.3.2. To make the implementations as comparable as possible, we applied the same solver to the different optimization problems, namely the `fmincon` function implemented in the optimization toolbox of MATLAB. This function provides a method for

constrained nonlinear optimization based on sequential quadratic programming. In terms of classification accuracy, the different implementations of choquistic regression should perform exactly the same, at least theoretically, because they seek to maximize the same likelihood function under different but equivalent constraints. Practically, of course, different formulations of the optimization problem will yield slightly different solutions, although these differences should be small. This expectation is confirmed by the result of a 5-fold cross validation, which is summarized in Table 7.5; this table also shows results with standard logistic regression (LR) as a baseline.

data set	CR-orig	CR-AI	CR-AII	LR
ESL	.0655±.0255	.0668±.0227	.0639±.0208	.0678±.0255
ERA	.2908±.0312	.2880±.0292	.2907±.0312	.2873±.0275
LEV	.1478±.0202	.1491±.0222	.1530±.0213	.1686±.0240
MMG	.1685±.0240	.1697±.0232	.1661±.0232	.1712±.0268
CPU	.0241±.0223	.0244±.0197	.0196±.0236	.0672±.0346
CEV	.0743±.0127	.0835±.0120	.0726±.0135	.1382±.0170
BCC	.3041±.0581	.2840±.0556	.3065±.0524	.3079±.0586
DBS	.1413±.0715	.1330±.0648	.1130±.0645	.1472±.0573
MPG	.0663±.0244	.0644±.0281	.0636±.0254	.0627±.0277
SWD	.2186±.0187	.2169±.0276	.2143±.0225	.2202±.0244

Table 7.5: Classification accuracy for 2-additive choquistic regression respect to different methods (mean ± standard derivation derived from 10 repeats of 5-fold cross-validation).

Binary Classification by the Choquet kernel (Evaluation & Results)

- Finally, the last experiment for binary classification has been conducted in a kernel framework. First of all, the comparison has been accomplished between the Choquet kernel, polynomial kernel ($d = 1, 2, 3$) and RBF kernel. In this case, the setting was as follows:

  To measure the performance of the approach, conventional $0/1$ loss is used. In addition the experimental setup randomly splits the data into two parts, 80% for training and 20% for testing. The model which induced from training data is then evaluated on the testing data. This procedure is repeated 100 times, and the results are averaged. The C-parameter, namely the trade-off parameter has been chosen among $\{10^{-5}, \ldots, 10^{5}\}$ with step 10. The width parameter of the Gaussian kernel was selected from $\{10^{-4}, \ldots, 10^{0}\}$ with

step 10. Both parameters were selected by internal nested cross validation. The results overview is given by Table 7.6. For the methods we considered the polynomial kernels for $d = 1, 2, 3$, RBF kernel, Choquet kernel for $k = 1, 2, 3, n$ and Choquet kernel modification and as well the original Choquet integral underlying SVM method.

A summary in terms of pairwise win statistics is provided in Tables 7.7. As can be seen, CK k=n+MM compares quite favorably with the other approaches, especially with the other kernel methods, in terms of the 0/1 loss. Note that in this case, in order to make the correction the heuristic approach (convex combination) is taken into account. The Friedman test finds significant differences among of the eight methods (at the 1 % significance level).

7.5 Experimental Results Related to Measure Correction

In section 6.4, the approaches to find the closest monotone measure for an arbitrary measure have been widely discussed. Here the results for two methods, namely Pardalos method [83] and convex combination method, are presented. We used maximum likelihood and kernel based approaches to learn the optimal parameters (setting in (6.25) and (6.26)). As discussed, the Pardalos method gives the optimal solution under an L_2 norm, which in our case has computational complexity of $\mathcal{O}(2^m \log 2^m)$. Additionally the convex combination provides the approximate solution, at computational complexity cost of $\mathcal{O}(2^m)$. Tables 7.8 and 7.9 show these comparisons in terms of performance and distance. The results also confirm the above fact; IRT-BIN provides the optimal solution. In terms of performance, in the case of Choquet kernel there is no significant gain for IRT-BIN although the corresponding results in terms of distance are slightly closer to the original measure. In the case of maximum likelihood the results are slightly different. Here the reason is that for maximum likelihood approach there is no way to control the capacity, whereas due to the SRM idea, the goal is always to find a trade-off between the quality of fit and complexity of classifier. Thereforethe overfitting problem is expected to occur during learning. Specifically in the DenBosch data this can be seen obviously. To measure the performance, the conventional $0/1$ loss is used. The experimental setup is as follows: the data is randomly split into two parts; 80% for training and 20% for testing. The model which induced from the training data, is then evaluated on the testing data. This procedure is repeated 100 times, and the

data set	PLY d=1	PLY d=2	PLY d=3	CK k=2	CK k=3	CK K=n + MC	CI + SVM	RBF
ESL	.1100±.0585(8)	.0505±.0206(2)	.0546±.0154(5)	.0495±.0208(**1**)	.0531±.0251(4)	.0510±.0191(3)	.0671±.0252(6)	.0711±.0278(7)
ERA	.2952±.0243(5)	.3052±.0280(8)	.2892±.0347(3)	.2869±.0271(1)	.2961±.0255(6)	.2900±.0383(4)	.3005 ±.0263(7)	.2887±.0332(2)
LEV	.1638±.0259(7)	.1563±.0291(6)	.1475±.0292(3)	.1698±.0190(8)	.1503±.0292(4)	.1461±.0208(1)	.1464 ±.0244(2)	.1512±.0233(5)
MMG	.1715±.0234(8)	.1612±.0215(5)	.1664±.0339(7)	.1594±.0360(2)	.1600±.0313(3)	.1608±.0225(4)	.1585±.0162(**1**)	.1636±.0175(6)
CPU	.0797±.0417(8)	.0573±.0435(4)	.0768±.0390(7)	.0569±.0407(3)	.0598±.0331(5)	.0451±.0402(**1**)	.0488±.0501(2)	.0732±.0349(6)
CEV	.1601±.0099(8)	.0478±.0081(5)	.0305±.0096(**1**)	.0533±.0133(6)	.0763±.0180(7)	.0416±.0093(4)	.0386±.0100(3)	.0335±.0087(2)
BCC	.3091±.0257(8)	.2736±.0612(5)	.2655±.0544(2)	.2582±.0588(**1**)	.2691±.0400(3)	.2809±.0499(6)	.2709±.0570(4)	.2867±.0507(7)
DBS	.1409±.0700(8)	.1322±.0654(6)	.1217±.0572(3)	.1348±.0571(7)	.1174±.0667(2)	.1130±.0711(**1**)	.1304±.0435(5)	.1283±.0498(4)
MPG	.0904±.0287(4)	.0917±.0261(5)	.0936±.0336(7)	.0901±.0334(3)	.0932±.0278(6)	.0897±.0380(2)	.0801±.0203(**1**)	.0962±.0347(8)
average rank	**7.11**	**5.11**	**4.22**	**3.55**	**4.44**	**2.88**	**3.44**	**5.22**

Table 7.6: Average errors ± standard deviation

	PLY d=1	PLY d=2	PLY d=3	CK k=2	CK k=3	CK k=n + MM	CI + SVM	RBF
PLY d=1	—	2	1	1	2	0	1	1
PLY d=2	7	—	4	3	3	2	1	5
PLY d=3	8	5	—	3	4	3	5	6
CK k=2	8	6	6	—	7	4	3	6
CK k=3	7	6	5	2	—	2	4	7
CK k=n + MM	9	7	6	5	7	—	5	7
CI +SVM	8	8	4	6	5	4	—	6
RBF	8	4	3	3	2	2	3	—

Table 7.7: Win statistics (number of data sets on which the first method was better than the second one) for 80% training data for 0/1 loss case.

results are averaged. The C-parameter, namely the trade-off parameter has been chosen among $\{10^{-5}, \ldots, 10^{5}\}$ with step 10 by internal nested cross validation.

data set	Orig. Err.	CK + CC - Err.	CK + IRT-BIN Err.	CR* + CC - Err.	CR* + IRT-BIN Err.
ESL	.0510±.0191	.0510±.0191	.0510±.0191	.0794 ± .0361	.0728 ± .0266
ERA	.2975±.0237	.2905±.0292	.2905±.0292	.2868 ± .0300	.2865 ± .0306
LEV	.1467±.0209	.1503±.0348	.1503±.0348	.1447 ± .0319	.1463 ± .0285
MMG	.1648±.0316	.1600±.0276	.1648±.0279	.1915± .0412	.2297 ± .0457
CPU	.0601±.0326	.0621±.0514	.0592±.0575	.0768± .0748	.0988 ± .0753
CEV	.0484±.0250	.0438±.0311	.0451±.0302	.0603± .0181	.0553 ± .0088
BCC	.2703±.0404	.2491±.0478	.2491±.0478	.3555± .0539	.3082 ± .0491
DBS	.1232±.0656	.1348±.0694	.1435±.0768	.4233± .0532	.4167 ± .0567
MPG	.0726±.0335	.0718±.0260	.0692±.0269	.0779± .0300	.0810 ± .0471

Table 7.8: The comparison results for two different approaches for fuzzy measure correction for $0/1$ loss.

data set	L_2(Orig, CK + CC)	L_2(Orig, CK + IRT-BIN)	L_2(Orig, CR* + CC)	L_2(Orig, CR* + IRT-BIN)
ESL	0	0	.4868	.4847
ERA	2.3549	2.3325	.3935	.3990
LEV	.6080	.6080	.6738	.2974
MMG	6.8476	6.4883	5.6876	5.4539
CPU	11.7060	11.6679	2.0853	2.0518
CEV	3.0984	3.0883	2.7439	2.2647
BCC	22.5812	21.5944	31.8797	27.3302
DBS	9.7976	9.1888	53.3094	49.1792
MPG	3.1719	2.9410	8.6131	8.4596

Table 7.9: The comparison results for two different approaches for fuzzy measure correction in terms of L_2 distance.

7.6 Experimental Results Regarding Ordinal Class Classification

Ordinal Classification by Ordinal Choquistic Regression (Evaluation & Results)

In the case of ordinal classification, the ordinal choquistic regression (OCR) was compared with ordinal logistic regression (OLR). As mentioned before, it is expected, that ordinal choquistic regression, due to its flexibility, namely the ability to capture nonlinear dependencies between predictor values and response, can improve the accuracy. Additionally in order to decrease the effect of overfitting, the OCR version with hierarchal regularization is equipped, which in Table 7.10 is shown as OCR+R. The experiments for ordinal choquistic regression referred to in Section 4.2.3, have been performed as follows: the data is randomly is split into two parts, one half for training and one half for testing. The model induced from training data is then evaluated on the test data. In order to measure performance, the L_1 loss was taken into account. This procedure is repeated 100 times, and the results are averaged.
The function $f(\cdot)$ in the regularization term was defined as $f(k) = k^\alpha$. Thus two hyper-parameters need to be tuned for OCR+R, namely ρ and α. This tuning was done by searching the grid

$$(\alpha, \rho) \in \Big\{2, 4, 6, 8\Big\} \times \Big\{10^{-4}, 10^{-3}, \cdots, 10^{4}\Big\}$$

and evaluating parameter combinations by means of a (nested) cross validation of the training data. Table 7.10 provides a summary of the results in terms of average L_1 loss. As can be seen, OCR often achieves clear improvements over OLR, especially in those data sets for which the response is known to depend on the predictors in a non-linear way. Moreover, our regularization method has payed off, as well, since the results of OCR+R are often even better than those of OCR. Once a choquistic model has been learned on a given set of training data, it can be used to predict the class of a new query instance $\boldsymbol{x} \in \mathcal{X}$. This prediction, however, is not straightforward, since does not produce a class prediction directly. Instead, it maps $\boldsymbol{x}$ to a probability distribution

$$\Big(\mathbf{P}(y_1 \mid \boldsymbol{x}), \ldots, \mathbf{P}(y_k \mid \boldsymbol{x})\Big) \in \Big[0, 1\Big]^{\mathcal{Y}}$$

from which a class prediction has to be derived. The most obvious prediction, of course, is the mode of this distribution:

$$\hat{y} = \arg\max \Big(\mathbf{P}(y_1 \mid \boldsymbol{x}), \ldots, \mathbf{P}(y_k \mid \boldsymbol{x}) \Big) \tag{7.2}$$

Indeed, this prediction minimizes the risk with respect to the $0/1$ loss. The risk minimizer with respect to the L_1 loss, however, is the median of the distribution:

$$\hat{y} = \arg \operatorname{med} \Big(\mathbf{P}(y_1 \mid \boldsymbol{x}), \ldots, \mathbf{P}(y_k \mid \boldsymbol{x}) \Big) \tag{7.3}$$

data set	OLR	OCR	OCR + R
ESL	.3094±.0325(1)	.3504±0.0939(3)	.3361±.0427(2)
ERA	1.2520±.0393(1)	1.2770±0.0279(3)	1.2617±.0292(2)
LEV	.4264±.0148(3)	.4184±0.0187(1)	.4224±.0242(2)
CEV	.2310±.0075(3)	.1097±.0361(1)	.1097±.0361(1)
MPG	.3648±.0324(1)	.3916±0.0349(2)	.4005±.0438(3)
CYD-1	.3167±.0441(3)	.1778±0.0536(2)	.1611±.0509(1)
CYD-2	.7722±.0712(3)	.3500±0.0810(2)	.3472±.0885(1)
CYD-3	.4667±.0471(3)	.2722±0.0360(2)	.2694±.0386(1)
CYD-4	.5133±.0414(3)	.2833±0.0583(2)	.2783±.0634(1)
CYD-5	.3100±.0465(3)	.2633±0.0477(2)	.2500±.0373(1)
CYD-6	.5083±.0874(3)	.2556±0.0750(2)	.2500±.0667(1)
CYd-7	.7150±.0541(3)	.3867±0.0628(2)	.3850±.0739(1)
average rank	**2.5**	2	**1.41**
ESL	.3400±.0504(1)	.3488±.0464(3)	.3456± .0184(2)
ERA	1.2824±.0648(2)	1.292±.0552(3)	1.2712±.0384(1)
LEV	.4372±.0344(3)	.4164±.0140(1)	.4204±.0148(2)
CEV	.2205±.0096(3)	.1203±.0291(2)	.1137±.0246(1)
MPG	.3365±.0375(3)	.3105±.0335(2)	.3045±.0310(1)
CYD-1	.3479±.0490(3)	.1952±.0498(2)	.1896±.0493(1)
CYD-2	.8167±.1017(3)	.3483±.0644(2)	.3425±.0698(1)
CYD-3	.4167±.0786(3)	.2700±.0375(1)	.2733±.0425(2)
CYD-4	.4633±.0576(3)	.3000±.0437(2)	.2933±.0432(1)
CYD-5	.3067±.0562(3)	.2833±.0360(2)	.2724±.0410(1)
CYD-6	.5583±.0748(3)	.2867±.0461(2)	.2783±.0409(1)
CYD-7	.7711±.0727(3)	.3289±.0682(1)	.3380±.0610(2)
average rank	**2.75**	1.91	**1.33**

Table 7.10: Average L_1 loss $\pm$ standard deviation (in brackets the rank). The results above refer to the median predictor (7.3), the results below to the model predictor(7.2).

The critical distance of ranks in the Nemenyi test ($\alpha = .10$) is .82 for the $0/1$ measure. In Table 7.3, the average ranks for which this difference is exceeded are highlighted in bold font.

7.7 Experimental Results for Complexity Reduction

The core idea of exploiting the correlation issue and ignoring high correlated features has been discussed in Subsection 6.3.1. In order to show how efficient the complexity reduction method is, some experiments on datasets were undertaken. In our setup the algorithm tries to find the smallest k-additivity at which the $\epsilon - \delta$ property can hold.

Table 7.11: Performance in terms the average Error $\pm$ standard deviation for dimensionality reduction case ($\epsilon = \delta = .1$).

	dataset	mode of selected k's	0/1 loss
20%	ESL	3	.0737±.0103
	ERA	4	.2981±.0158
	LEV	4	.1526±.0146
	CPU	4	.0998±.0347
	MMG	3	.1761±.0107
	CEV	6	.0448±.0089
	BCC	3	.2888±.0578
	DBS	4	.2286±.0549
	MPG	4	.0719±.0108
50%	ESL	3	.0727±.0148
	ERA	4	.2930±.0162
	LEV	4	.1421±.0142
	CPU	4	.0361±.0432
	MMG	3	.1667±.0130
	CEV	6	.0376±.0059
	BCC	3	.2838±.0448
	DBS	4	.1944±.0631
	MPG	4	.0570±.0080
80%	ESL	3	.0603±.0236
	ERA	4	.2899±.0191
	LEV	4	.1370±.0162
	CPU	4	.0244±.0531
	MMG	3	.1620±.0250
	CEV	6	.0273±.0089
	BCC	3	.2755±.0404
	DBS	4	.1939±.0615
	MPG	4	.0597±.0126

More precisely, the algorithm starts with the highest k, in this case k = number of attributes, and if the $\epsilon - \delta$ property can be held for k, the algorithm tries it

recursively for $k-1$. If the algorithm cannot hold, then the $\epsilon-\delta$ property algorithm stops, and as an output indicates the smallest k in terms of the $\epsilon-\delta$ property. In our experiments, we used $\epsilon=\delta=.1$ and for five datasets the algorithm can apparently reduce the complexity (7.11). Needless to say, that using different $\epsilon-\delta$ we could reduce the complexity for other datasets. Note that the algorithm to find the k^* was proposed in Subsection 6.3.1.

7.8 Experimental Results with Respect to Running Time

The Choquet integral in general, and the Möbius transform in particular an exponential number of constraints are needed for assuring the monotonicity issue. Learning this kind of model for such a high number of attributes is indeed a challenging problem. To overcome the complexity issue, several algorithms have been proposed in chapter 6. Also, several experiments were performed to exhibit the advantages of the proposed algorithms in chapter 6. In the following discussion, the corresponding results in terms of running time are presented.

7.8.1 2-additive Choquet Integral

Experimentally, three different versions of the choquistic regression proposed in 6.3.2 have been compared, the original formulation (CR-orig), the first reformulation (CR-I) and the second reformulation (CR-II) (See Subsection 6.3.2). To make the implementations as comparable as possible, the same solver to the different optimization problems was applied, namely the `fmincon` function implemented in the optimization toolbox of MATLAB. This method is based on a sequential quadratic programming approach. In terms of classification accuracy, the different implementations of choquistic regression should perform in exactly the same manner, at least theoretically, because they seek to maximize the same likelihood function under different, but equivalent constraints. Practically, of course, different formulations of the optimization problem will yield slightly different solutions, although these differences should be small. This expectation is confirmed by the result of a 5-fold cross validation, which is summarized in Table 7.5.
What we are of course most interested in is the runtime performance of the different implementations, which we measured in terms of CPU usage [1]. The results, which are summarized in Table 7.12, convey a quite clear picture: While the original implementation CR-orig is superior, or at least competitive, for data sets with up to

[1] Experiments were carried out on an Intel Core(TM) i7-2600 CPU with 3.40GHz and 8 GB RAM under Windows 7.

6 attributes, it is visibly outperformed by the alternative formulations for $m > 6$ attributes, and the difference in runtime rapidly increases with m.

data set	method	20%	40%	60%	80%	100%
ESL	CR-orig	0.26 ± 0.05	0.31 ± 0.02	0.38 ± 0.02	0.45 ± 0.13	0.63 ± 0.05
	CR-I	0.41 ± 0.13	0.50 ± 0.07	0.68 ± 0.13	0.80 ± 0.17	1.05 ± 0.18
	CR-II	0.31 ± 0.09	0.39 ± 0.07	0.50 ± 0.06	0.61 ± 0.04	0.70 ± 0.04
ERA	CR-orig	0.2312 ± 0.03	0.3699 ± 0.01	0.5093 ± 0.02	0.6366 ± 0.01	0.7812 ± 0.02
	CR-I	0.5368 ± 0.10	0.9053 ± 0.08	1.068 ± 0.16	1.2012 ± 0.20	1.3559 ± 0.18
	CR-II	0.3181 ± 0.05	0.5273 ± 0.07	0.7097 ± 0.09	1.1253 ± 0.14	1.3255 ± 0.16
LEV	CR-orig	0.34 ± 0.04	0.55 ± 0.05	0.71 ± 0.04	0.88 ± 0.07	1.03 ± 0.07
	CR-I	0.96 ± 0.23	1.41 ± 0.21	1.84 ± 0.24	2.25 ± 0.18	2.5 ± 0.19
	CR-II	0.49 ± 0.07	0.76 ± 0.05	1.04 ± 0.10	1.68 ± 0.15	1.90 ± 0.14
MMG	CR-orig	0.39 ± 0.15	0.56 ± 0.06	0.79 ± 0.12	0.95 ± 0.09	1.07 ± 0.11
	CR-I	1.19 ± 0.24	1.77 ± 0.47	2.06 ± 0.61	2.71 ± 1.60	3.24 ± 1.96
	CR-II	0.52 ± 0.13	0.83 ± 0.11	1.13 ± 0.10	1.54 ± 0.18	1.78 ± 0.19
CPU	CR-orig	0.77 ± 0.18	1.95 ± 3.39	3.37 ± 5.42	6.9 ± 8.97	14.23 ± 11.33
	CR-I	1.85 ± 0.22	2.56 ± 0.52	2.79 ± 0.71	3.42 ± 0.18	6.11 ± 2.71
	CR-II	0.50 ± 0.31	1.28 ± 0.24	1.33 ± 0.29	1.68 ± 0.56	2.06 ± 0.66
CEV	CR-orig	2.45 ± 0.24	3.84 ± 0.38	5.09 ± 0.41	5.79 ± 0.51	6.74 ± 0.41
	CR-I	5.36 ± 0.55	7.53 ± 1.00	9.89 ± 0.96	11.93 ± 2.83	13.72 ± 2.56
	CR-II	2.11 ± 0.33	3.68 ± 0.31	5.23 ± 0.52	6.88 ± 0.59	7.88 ± 0.58
BCC	CR-orig	1.22 ± 0.56	1.10 ± 0.27	1.19 ± 0.23	1.47 ± 0.38	1.47 ± 0.25
	CR-I	2.29 ± 1.09	2.04 ± 1.52	2.16 ± 0.95	2.88 ± 2.5	2.97 ± 2.3
	CR-II	0.47 ± 0.24	0.47 ± 0.06	0.55 ± 0.55	0.66 ± 0.11	0.78 ± 0.07
DBS	CR-orig	5.68 ± 1.11	5.36 ± 1.23	5.61 ± 1.02	5.59 ± 0.72	5.47 ± 1.05
	CR-I	2.51 ± 1.81	2.88 ± 1.29	3.03 ± 1.42	3.17 ± 0.96	4.08 ± 1.10
	CR-II	0.71 ± 0.19	0.78 ± 0.34	0.76 ± 0.18	0.82 ± 0.12	0.91 ± 0.13
MPG	CR-orig	1.83 ± 0.71	2.15 ± 0.62	2.69 ± 0.59	3.18 ± 0.54	3.45 ± 0.65
	CR-I	2.58 ± 0.32	2.54 ± 0.66	3.46 ± 0.89	3.84 ± 0.75	4.15 ± 0.92
	CR-II	0.61 ± 0.21	0.72 ± 0.12	0.95 ± 0.24	1.02 ± 0.19	1.3 ± 0.13
SWD	CR-orig	292.41 ± 31.11	382.82 ± 42.24	371.32 ± 12.67	394.00 ± 36.62	427.54 ± 36.62
	CR-I	17.92 ± 13.43	27.82 ± 12.13	32.11 ± 10.10	32.35 ± 10.05	33.14 ± 10.77
	CR-II	4.71 ± 0.71	8.80 ± 1.34	13.01 ± 1.44	18.24 ± 2.21	22.66 ± 1.73

Table 7.12: Runtime complexity of the different methods measured in terms of CPU time (mean ± standard deviation) for different sample sizes (in % of the complete data set).

This is in agreement with our expectations: An exponential number of constraints is no big obstacle provided the number of attributes is small. In this case, a reduction from exponential to quadratic does not compensate for the additional overhead caused by introducing new variables. Due to the exponential growth of the number of constraints in CRorig, however, this situation quickly changes in favor of CR-AI and CR-AII with an increasing number of attributes; indeed, as can

be seen from the SWD data, the runtime of CR-orig becomes unacceptable as soon as $m > 9$. This is also confirmed by another experiment we did with this data set: From the total of 10 attributes, we randomly sampled $m \in \{5, 6, ..., 10\}$, trained a CR model on the data set reduced to these k attributes (using the tree methods CR-orig, CRAI and CR-AII) and measured the runtime. This was repeated 100 times and the runtime was averaged. Figure 7.1 shows this average runtime as a function of m. Comparing the two alternatives CR-AI and CR-AII, it seems that the latter is consistently faster, although the growth of the runtime as a function of m is in both cases much more moderate than for CR-orig. Again, this is not unexpected against the background of the results from the previous section.

Our experimental results are in complete agreement with the theoretical complexity (in terms of the number of constraints and the number of variables involved) of the optimization problems. Thus, learning the Choquet integral for classification can indeed be made more efficient by exploiting the special structure of the problem in the case of 2-additive fuzzy measures, essentially reducing the complexity from exponential to quadratic in the number of attributes. In order to compare the different variants of the problem (CR-orig, CR-AI, CRAII), we decided to use a rather general optimization method that can handle all of them without the need for specific adaptations. An interesting alternative, of course, is to implement each of the variants individually and as efficiently as possible, seeking for a more specialized solver that allows for exploiting the respective problem structure in an optimal way. In particular, this appears to be important for a more thorough comparison of the two alternatives we proposed, respectively, in Subsections 6.3.2. Theoretically, CR-AII seems to be advantageous to CR-AI, and indeed, the experimental results are in agreement with this presumption. Nevertheless, the reformulation in Section 6.3.2 should not be abandoned rashly. First, as just mentioned, it might be possible to improve its efficiency by means of specialized optimization techniques; one may think, for example, of an alternating optimization scheme in which, repeatedly, the $\alpha_{i,j}$ are fixed while the $m_{i,j}$ are optimized and vice versa, thereby circumventing the issue of nonlinearity. Moreover, CR-AII might be more amenable for a generalization to the case of k-additive measures, $k > 2$. In this regard, the second approach is arguably difficult: Firstly, it is known that for k > 2, the extreme points of the convex polytope of k-additive measures are not all $\{0, 1\}$-valued. Secondly, and more importantly, the number of these extreme points is expected to grow extremely quickly, knowing that the number of extreme points of the polytope of additive measures on m variables grows like the sequence of Dedekind numbers [74].

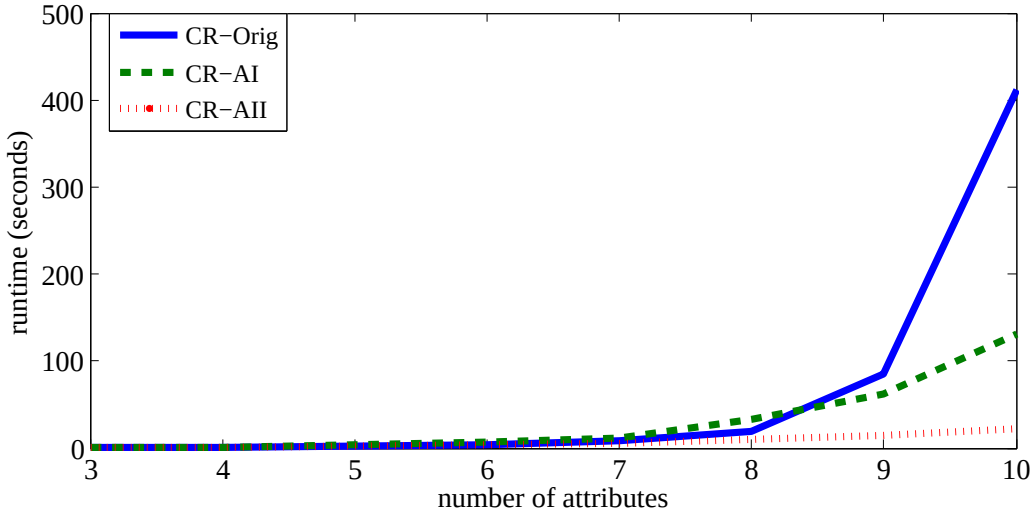

Figure 7.1: Average runtime on the SDW data as a function of the number of attributes included

7.8.2 The Choquet Kernel

The compact form of kernel representation of the Choquet integral exhibits more advantages, especially in terms of complexity reduction. Hence it is expected that the kernel representation, specifically for binary classification purpose, has a lower run time compared to the original form. To this end, several experiments have been conducted to show this fact. An overview of results are demonstrated in Figure 7.2 and correspond to $CKk = n+MC$. In fact, the run time involves the measure modification part (convex combination) to assure monotonicity. The results recommend that if the number of attributes are small while the number of training examples are high the primal setting has an advantage, whereas by having a high number of attributes and low number of training examples our proposal is preferred. However there is no exact definition for such a comparison. As it can be seen, for the data sets CPU, DenBosch and CarMPG the compact representation of the Choquet kernel with a dual form is dramatically faster than the original form. As discussed, these data sets meet the mentioned property, i.e., here the number of attributes are fairly large, whereas the number of instances are low.

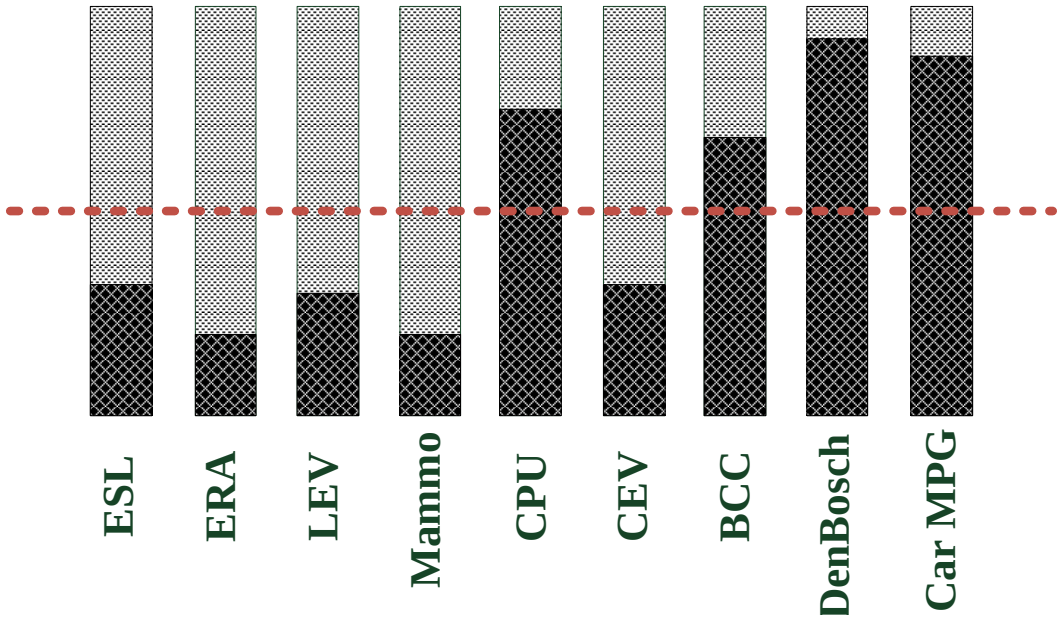

Figure 7.2: Illustration of run time with respect to primal and dual setting for different data sets. The dark parts correspond to the fraction of run time for primal setting to the run time of primal setting plus run time of dual setting with our proposed setting for each data sets individually. The dash line cuts the rectangles in the middle, which means if the dark part ends on the dash line both methods would have the same run time.

7.9 Interpretation and Illustration

One of the key features of our approaches, which we already mentioned in Section 3.6 is the aspect of interpretability. In particular, the Choquet integral (or, more specifically, the underlying fuzzy measure) provides natural measures of the importance of individual attributes and the interaction between pairs (or even groups) of attributes [45]. In fact, in many practical applications, this type of information is at least as important as the prediction accuracy of the model. These information can be exploited to understand the model better. In the following discussion, we show some real cases regarding the Shapley index and interaction index.

Shapley Index

As described earlier, the Shapley index measures the importancy of each criterion (attribute) and such information can be used to interpret the model. In many practical applications, this type of information is at least as important as the prediction

accuracy of the model. So here we just give a few examples showing the plausibility of the results. The results are related to the choquistic regression model (binary classification case).

Regarding the Shapley index, the (average) values on the Car MPG data are as follows:

$$\begin{gathered}\text{cylinders} \approx 0.13 \\ \text{displacement} \approx 0 \\ \text{horsepower} \approx 0.25 \\ \text{weight} \approx 0.46 \\ \text{acceleration} \approx 0.03 \\ \text{model year} \approx 0.13 \\ \text{origin} \approx 0\end{gathered}$$

In terms of attribute importance, this conveys the following picture:

$$\begin{aligned} I(\text{weight}) \succ I(\text{horsepower}) \succ I(\text{cylinders}) | I(\text{ model year}) \\ \succ I(\text{acceleration}) \succ I(\text{displacement}) | I(\text{origin}) \end{aligned}$$

Recalling the meaning of the data set, these weights should reflect the influence on the fuel consumption, and seen from this point of view, they appear to be fully plausible.

For the CPU data, the following Shapley values are obtained:

$$\begin{gathered}\text{machine cycle time in nanoseconds} \approx 0.07 \\ \text{minimum main memory in kilobytes} \approx 0.24 \\ \text{maximum main memory in kilobytes} \approx 0.30 \\ \text{cache memory in kilobytes} \approx 0.20 \\ \text{minimum channels in units} \approx 0.10 \\ \text{maximum channels in units} \approx 0.09\end{gathered}$$

Thus, the most important properties are those concerning the memory (main and cache). The influence of the other properties (channels, cycle time) is not as strong, although they are not completely unimportant either.

Interaction Index

Apart from the importance of individual attributes, it is interesting to look at the interaction between different attributes. A detailed analysis of this type of information is difficult and beyond the scope of this paper. Yet, just to give an example as an illustration, 7.3 visualizes the (pairwise) interaction between attributes for the car evaluation data, for which CI performs significantly better than WM. Recall that, in this data set, the evaluation of a car (output attribute) depends on a number of criteria, namely (a) buying price, (b) price of the maintenance, (c) number of doors, (d) capacity in terms of persons to carry, (e) size of the luggage boot, and (f) safety of the car. These criteria form a natural hierarchy: (a) and (b) form a subgroup PRICE, whereas the other properties are of a TECHNICAL nature and can be further decomposed into COMFORT (c–e) and SAFETY (f). Interestingly, the interaction in our model nicely agrees with this hierarchy: Interaction within each subgroup tends to be smaller (as can be seen from the darker colors) than interaction between criteria from different subgroups, suggesting a kind of redundancy in the former and complementarity in the latter case.

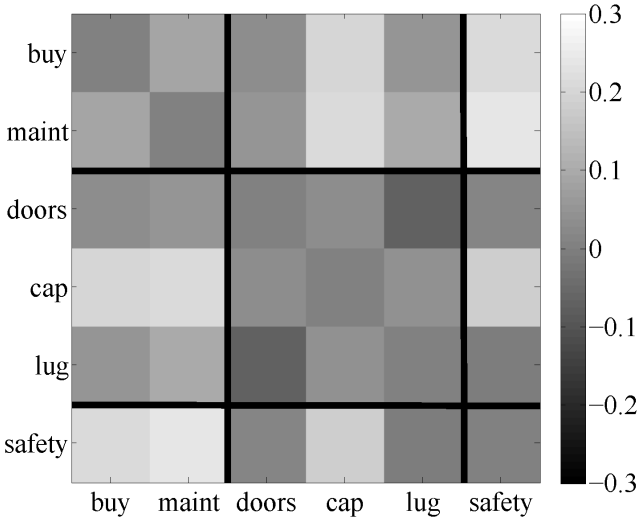

Figure 7.3: Visualization of the interaction index for the car evaluation data (numerical values are shown in terms of level of gray, values on the diagonal are set to 0). Groups of related criteria are indicated by the black lines.

In addition, Figure (7.4) visualizes the interaction between the three attributes in the color yield data sets, namely for CLR-1 and CLR-7. Degrees of interaction are shown as levels of gray, which means that light and dark fields strongly silhouetted against the color of the diagonal indicate a high degree of interaction. Obviously, the interaction is not very strong in the case of CYD-1, but more pronounced for

CYD-7. This is in agreement with the improvement in terms of accuracy, which is much higher in the latter case.

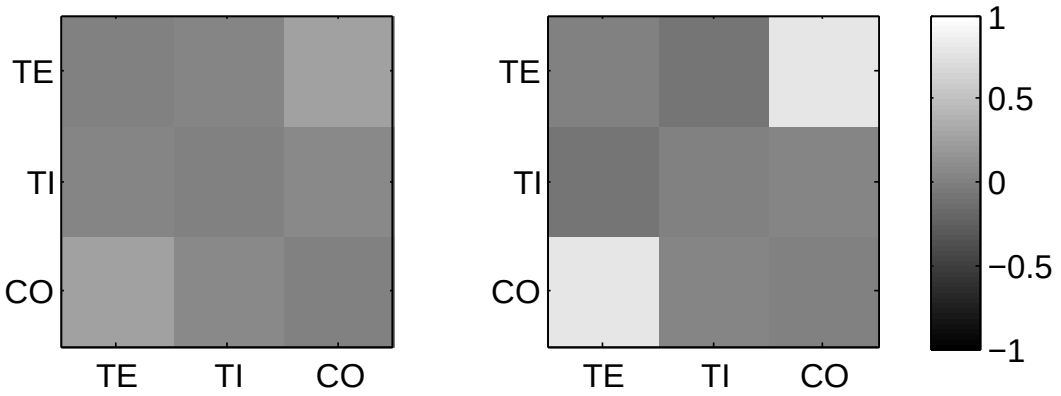

Figure 7.4: Visualization of the interaction index. In terms of the 0-1 loss, the improvement brought by the Choquet integral over the weighted sum is 0.067 on the left and 0.133 on the right (TE: temperature, TI: time, CO: concentration).

Illustration of PCA Kernel

From a k-additivity point of view, we conducted one experiment to demonstrate the advantages of different levels of complexity regarding the k-additive Choquet kernel. In this case, we plotted the scatter plot for the data set DenBosch for different levels of the Choquet kernels. As can be seen in Figure (7.5), a lower order of the Choquet kernel ($k = 2$) tends to produce overlapping data regions for the two classes, whereas by using a higher order of the Choquet kernel, namely $k = 3$ or even the complete attribute dependency structure for $k = 8$ an improvement obviously exists. In other words, for higher orders of the Choquet kernel, there is an opportunity to separate two classes more precisely. Note that, this observation does not imply that for higher orders of the Choquet kernel, a sound improvement always exists. Of course this depends on the data from the optimal setting with respect to the complexity of kernel chosen.

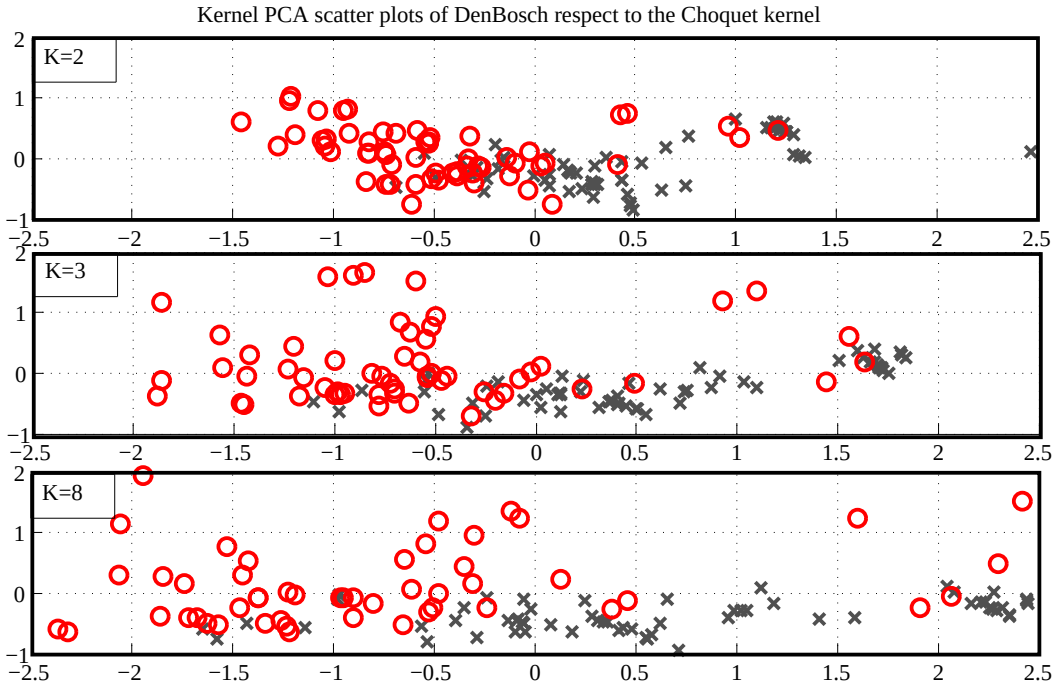

Figure 7.5: Illustrative scatterplot visualizations of the data under the Choquet kernels of different orders k are obtained by using kernel principal component analysis [88].

The Illustration of Monotonicity Constraints Satisfaction by Monotone Datasets

The Choquet kernel representation brings some advantages in terms of complexity. As mentioned earlier, the scenario is to adapt the Choquet kernel by given monotone data as training data in advance. Then the learned parameters should be modified by some modification methods. In this regard, the satisfaction of monotonicity constraints was measured, in the sense that the satisfied constraints was compared to the unsatisfied constraints. That is we computed the proportion of satisfied and unsatisfied constraints, respectively, to the whole number of monotonicity constraints. As is shown in Figure (7.6), expanding the number of training data can improve the satisfaction of monotonicity constraints in a meaningful way.

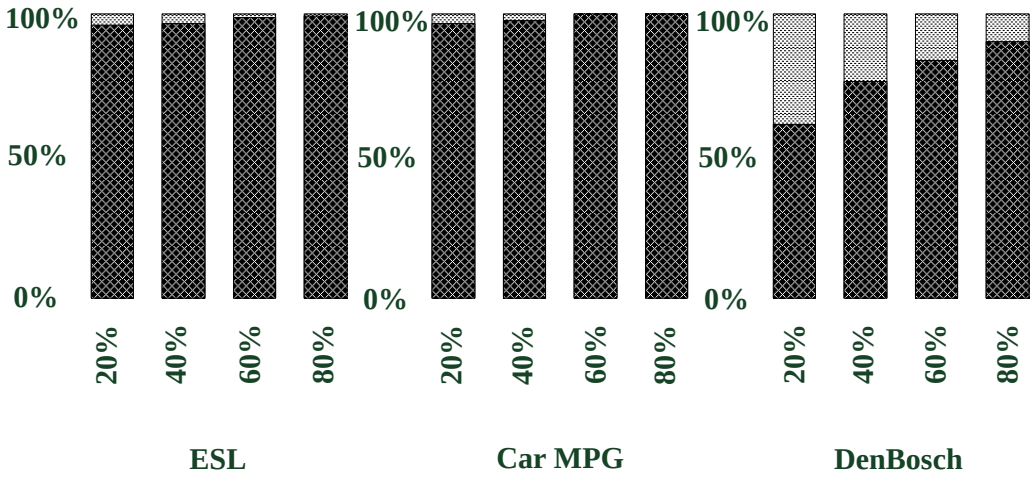

Figure 7.6: The illustration of satisfying monotonicity constraints by expanding the number of training data. It shows that adding the number of instances reduces the number of unsatisfied constraints.

Evaluation of Precision Parameter

As mentioned earlier, one may expect a close connection between the scaling parameter γ in the choquistic model and the prediction accuracy of the model. More specifically, the better the model performs on a particular data set, the higher γ it is expected to be. It is worth mentioning that our experimental results are in perfect agreement with this expectation. Indeed, comparing the ranking of the nine data sets in terms of accuracy and in terms of the average values of γ (shown in Table 7.13), we obtain a (Kendall tau) correlation of more than 0.8 throughout.

DBS	CPU	BCC	MPG	ESL	MMG	ERA	LEV	CEV
36.69	691.81	15.30	23.87	45.12	19.05	8.07	15.13	69.23

Table 7.13: Average values of the scaling parameter γ in the choquistic regression model.

8
Conclusion & Outlook

8.1 Conclusion

The learning of predictive models that guarantee a monotonic relationship between the output (response) and input (predictor) variables has received increasing attention in machine learning in recent years. [7, 10, 12, 14, 29, 33, 35, 60, 67, 68, 85, 98]. From a machine learning point of view, monotonicity usually is defined with respect to supervised learning, in which a kind of dependency between input attributes and output is taken into account. Roughly speaking, monotonicity means, the increase (decrease) of a certain input variable can only produce an increase in the output variable. Since such dependency is not necessarily linear, exploiting non-linear models which guarantee monotonicity provide benefits. From this perspective, this thesis focuses on non-linear models which also guarantee monotonicity.
Interestingly, monotonicity is not easily guaranteed for a number of well-known classification methods like, for example, decision trees. Thus, for a decision tree it may easily happen that, depending on the values of the remaining attributes, increasing the value of an attribute (e.g., tobacco consumption) may change the class prediction from positive to negative in one case (e.g., no heart attack) and from negative to positive in another case (e.g., heart attack). In this regard to use the so-called background knowledge, the parameters of the learner should be tuned under certain conditions. These conditions restrict the optimal solution to a sub-optimal solution,

in which the monotone model can be assured. Needless to say, such constraints are designed for each learner individually.

In this thesis we advocated the usefulness of the Choquet integral, specifically from a machine learning point of view. First in Chapter 3, the idea of MCDA and common approaches were discussed in general, whereas in particular the fuzzy integrals, especially the Choquet integral were presented. In order to employ the Choquet integral into a machine learning framework, several algorithms for (ordinal) classification were proposed. In Chapter 4, the algorithms for binary/ordinal classification were introduced and meanwhile the (ordinal) choquistic regression as a generalization of common (ordinal) linear regression was offered. In Chapter 5, the (ordinal) classification problem was considered from a kernel-based learning perspective as well. In this regard, the concept of the Choquet kernel was introduced. Since the learned parameters in the kernel framework do not obey monotonicity properties necessarily, we presented several approaches to repair such inconsistencies as well.
As discussed several times in thesis, there are always computational difficulties involved with estimating parameters for the Choquet integral, underlying the Möbius transform. This issue is addressed as a complexity issue. In Chapter 6, several algorithms were proposed to reduce the complexity of such models. Specifically several reductions for the case of a 2-additive Choquet integral case were presented. Additionally the Choquet integral underlying the fuzzy measure provides promising information about the interpretability of the model, namely, the index which measures the importancy of each criteria and the index which measures the dependency between criteria in terms of positive or negative synergies. In Chapter 7, the interpretability of the Choquet integral and the corresponding results for proposed algorithms were shown. In this regard, through some experiments, the reliability of such indicators was demonstrated.

8.2 Outlook

As discussed in the very beginning of the thesis, the main challenge for monotone learner is to enforce the monotonicity. Here the proposed models underlying the Choquet integral luckily ensure monotonicity, although at a cost of exponentially many constraints. However, in Chapter 6 the idea of learning without monotonicity constraints was presented, but one crucial question is, how it is possible to reduce the number of constraints? For the 2-additive Choquet integral, this issue was

widely discussed. In fact, for the general case such reductions make perfect sense. Such reductions are not only useful in terms of computational complexity but also make the interpretation simpler. These reductions are fully independent to the data, however one can imagine that sometimes data hints to remove some unnecessary weights. This issue was addressed in Subsection 6.3.1 as "Complexity Reduction by Exploiting Dependency". This method provides an upper bound of k-additivity. In an advanced version, the idea is to detect unnecessary weights and omit them. Indeed omitting the useless weights besides of lessen a computational complexity, can improve quality of fit, because certainly the risk of overfitting problem can be reduced.
In this thesis, a family of kernels underlying the Choquet integral was introduced, and several benefits of employing these kernels were shown. For the Choquet kernel is not any monotonicity constraint considered, in fact the kernel based learning only consider constraints with respect to instances and their classes. The number of these constraints corresponds to the number of training examples. The learning problem can be tackled by *kernelized logistic regression* too, where the information of classes and instances are involved in objective function. Indeed by kernelized logistic regression the learning procedure can be carried out without any constraint, which is perhaps, a way to reduce the complexity too. Hence from a computational complexity point of view there exists a benefit.

As one of useful information, that the proposed models provide is the information about joint weights of attributes, which are called interaction index. The interaction index are derived from a fuzzy measure, means, derived from learned weights. However another problem of interest is to estimate these values, or at least the sign of them, in advance. This could be done by considering dependencies between attributes (unsupervised learning). The estimation, even in basic phase, namely, sign of interactions, may then decrease the complexity (in terms of running time). In fact, any information derived from unsupervised learning, which can reduce the complexity in general, and omit the useless weights in particular is desirable.

Bibliography

[1] J. Aldrich. R. A. Fisher and the making of maximum likelihood 1912-1922. *Statistical Science*, 12(3):pp. 162–176, 1997.

[2] S. Angilella, S. Greco, and B. Matarazzo. Non-additive robust ordinal regression with Choquet integral, bipolar and level dependent Choquet integrals. In Joao Paulo Carvalho, Didier Dubois, Uzay Kaymak, and Joao Miguel da Costa Sousa, editors, *Proceedings of the Joint International Fuzzy Systems Association World Congress and European Society of Fuzzy Logic and Technology Conference*, pages 1194–1199. IFSA/EUSFLAT, 2009.

[3] S. Angilella, S. Greco, and B. Matarazzo. The most representative utility function for non-additive robust ordinal regression. In Eyke Hüllermeier, Rudolf Kruse, and Frank Hoffmann, editors, *Computational Intelligence for Knowledge-Based Systems Design*, volume 6178 of *Lecture Notes in Computer Science*, pages 220–229. Springer, 2010.

[4] S. Angilella, S. Greco, and B. Matarazzo. Non-additive robust ordinal regression: A multiple criteria decision model based on the Choquet integral. *European Journal of Operational Research*, 201:277–288, 2010.

[5] F. Bach. High-dimensional non-linear variable selection through hierarchical kernel learning. arXiv:0909.0844, September 2009.

[6] V. Balla, C. Gaganis, F. Pasiouras, and C. Zopounidis. Multicriteria decision aid models for the prediction of securities class actions: evidence from the banking sector. *OR Spectrum*, 36(1):57–72, 2014.

[7] N. Barile and A. J. Feelders. Nonparametric monotone classification with MOCA. In *ICDM*, pages 731–736. IEEE Computer Society, 2008.

[8] G. Beliakov and S. James. Citation-based journal ranks: the use of fuzzy measures. *Fuzzy Sets and Systems*, 167(1):101–119, March 2011.

[9] A. Ben-David. http://mldata.org/repository/data/viewslug/datasets-arie_ben_david-swd/.

[10] A. Ben-David. Monotonicity maintenance in information-theoretic machine learning algorithms. *Machine Learning*, 19:29–43, 1995.

[11] A. Ben-David, L. Sterling, and Y. H. Pao. Learning and classification of monotonic ordinal concepts. *Computational Intelligence*, 5(1):45–49, 1989.

[12] A. Ben-David, L. Sterling, and T. Tran. Adding monotonicity to learning algorithms may impair their accuracy. *Expert Systems with Applications*, 36(3):6627–6634, 2009.

[13] D. Bigot, H. Fargier, B. Zanuttini, and J. Mengin. Using and Learning GAI-Decompositions for Representing Ordinal Rankings. In Johannes Fürnkranz and Eyke Hüllermeier, editors, *ECAI'2012 workshop on Preference Learning (PL), Monpellier, 28/08/2012*, pages 5–10, http://www.ke.tu-darmstadt.de/events/PL-12/workshop.html, 2012. TU Darmstadt.

[14] J. C. Bioch and V. Popova. Monotone decision trees and noisy data. Research Paper ERS-2002-53-LIS, Erasmus Research Institute of Management (ERIM), ERIM is the joint research institute of the Rotterdam School of Management, Erasmus University and the Erasmus School of Economics (ESE) at Erasmus Uni, 2002.

[15] Ø. Birkenes. *A framework for speech recognition using logistic regression*. PhD thesis, Trondheim : Norwegian University of Science and Technology, Faculty of Information Technology, Mathematics and Electrical Engineering, Department of Electronics and Telecommunications, 2007.

[16] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[17] H. Block, S. Qian, and A. Sampson. Structure algorithms for partially ordered isotonic regression. *Journal of Computational and Graphical Statistics*, 3(3):285–300, 1994.

[18] J. P. Brans and B. Mareschal. PROMETHEE methods. In José Figueira, Salvatore Greco, and Matthias Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, chapter 5, pages 163 – 197. Springer's International Series, 2005.

[19] O. Burdakov, A. Grimwall, and M. Hussian. A generalised PAV algorithm for monotonic regression in several variables. In *COMPSTAT*, 2004.

[20] O. Burdakov, O. Sysoev, A. Grimvall, and M. Hussian. An $\mathcal{O}(n^2)$ algorithm for isotonic regression. In Gianni Pillo and Massimo Roma, editors, *Large-Scale Nonlinear Optimization*, volume 83 of *Nonconvex Optimization and Its Applications*, pages 25–33. Springer US, 2006.

[21] H. Bustince, J. Fernandez, and R. Mesiar. *Aggregation Functions in Theory and in Practise: Proceedings of the 7th International Summer School on Aggregation Operators at the Public University of Navarra, Pamplona, Spain, July 16-20, 2013*. Springer Publishing Company, Incorporated, 2013.

[22] R. Chandrasekaran, Y. Ryu, V. Jacob, and S. Hong. Isotonic separation. *INFORMS Journal on Computing*, 17:462–474, 2005.

[23] P. Y. Kwong Cheng. Improving financial decision making with unconscious thought: A transcendent model. *Journal of Behavioral Finance*, 11(2):92–102, 2010.

[24] G. Choquet. Theory of capacities. *Annales de l'institut Fourier*, 5:131–295, 1954.

[25] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

[26] H. Daniels and B. Kamp. Applications of MLP networks to bond rating and house pricing. *Neural Computation and Applications*, 8:226–234, 1999.

[27] K. De Brucker, C. Macharis, and A. Verbeke. Multi-criteria analysis in transport project evaluation: an institutional approach. *European Transport Trasporti Europei*, (47):3–24, 2011.

[28] K. Dembczyński, W. Kotłowski, and R. Słowiński. Additive preference model with piecewise linear components resulting from dominance-based rough set approximations. In *International Conference on Artificial Intelligence and Soft Computing 2006*, volume 4029 of *Lecture Notes in Computer Science*, pages 499–508, 2006.

[29] K. Dembczyński, W. Kotłowski, and R. Słowiński. Learning rule ensembles for ordinal classification with monotonicity constraints. *Fundamenta Informaticae*, 94(2):163–178, 2009.

[30] K. Dembczyński, G. Salvatore, W. Kotłowski, and R. Słowiński. Optimized generalized decision in dominance-based rough set approach. In JingTao Yao, Pawan Lingras, Wei-Zhi Wu, Marcin Szczuka, NickJ. Cercone, and Dominik Śl̈ezak, editors, *Rough Sets and Knowledge Technology*, volume 4481 of *Lecture Notes in Computer Science*, pages 118–125. Springer Berlin Heidelberg, 2007.

[31] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[32] D. Diakoulaki, C. H. Antunes, and A. Gomes Martins. MCDA and Energy Planning. In *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 78 of *International Series in Operations Research & Management Science*, pages 859–890. Springer New York, 2005.

[33] W. Duivesteijn and A. Feelders. Nearest neighbour classification with monotonicity constraints. In *Machine Learning and Knowledge Discovery in Databases*, volume 5211 of *Lecture Notes in Computer Science*, pages 301–316. Springer, 2008.

[34] J. S. Dyer. MAUT -Multi Atribute Utility Theory. In José Figueira, Salvatore Greco, and Matthias Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, chapter 7, pages 265 – 297. Springer's International Series, 2005.

[35] A. Feelders. Monotone relabeling in ordinal classification. In *Proceedings of the 10th IEEE International Conference on Data Mining*, pages 803–808. IEEE Computer Society, 2010.

[36] J. Figueira, S. Greco, and M. Ehrgott, editors. *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer's International Series, 2005.

[37] J. Figueira, V. Mousseau, and B. Roy. ELECTRE methods. In José Figueira, Salvatore Greco, and Matthias Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, chapter 4, pages 133 – 163. Springer's International Series, 2005.

[38] P. Flach. *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Cambridge University Press, New York, NY, USA, 2012.

[39] E. Frank and M. Hall. A simple approach to ordinal classification. In *Proc 12th European Conference on Machine Learning*, pages 145–156. Springer, 2001.

[40] K. Fukunaga and L. Hostetler. k-nearest-neighbor bayes-risk estimation. *Information Theory, IEEE Transactions on*, 21(3):285–293, 1975.

[41] F. Gebhardt. An algorithm for monotone regression with one or more independent variables. *Biometrika 57*, pages 263 – 271, 1970.

[42] M. Grabisch. Fuzzy integral in multicriteria decision making. *Fuzzy Sets and Systems*, 69(3):279–298, 1995.

[43] M. Grabisch. A new algorithm for identifying fuzzy measures and its application to pattern recognition. In *Proceedings of IEEE International Conference on Fuzzy Systems*, volume 1, pages 145–150. IEEE, 1995.

[44] M. Grabisch. The application of fuzzy integrals in multicriteria decision making. *European Journal of Operational Research*, 89(3):445 – 456, 1996.

[45] M. Grabisch. The representation of importance and interaction of features by fuzzy measures. *Pattern Recognition Letters*, 17(6):567–575, 1996.

[46] M. Grabisch. k-order additive discrete fuzzy measures and their representation. *Fuzzy Sets and Systems*, 92(2):167–189, 1997.

[47] M. Grabisch. Fuzzy integral for classification and feature extraction. In Michel Grabisch, Toshiaki Murofushi, and Michio Sugeno, editors, *Fuzzy Measures and Integrals: Theory and Applications*, pages 415–434. Physica, 2000.

[48] M. Grabisch. Modelling data by the Choquet integral. In *Information Fusion in Data Mining*, pages 135–148. Springer, 2003.

[49] M. Grabisch, I. Kojadinovic, and P. Meyer. A review of methods for capacity identification in Choquet integral based multi-attribute utility theory: Applications of the Kappalab R package. *European Journal of Operational Research*, 186(2):766–785, 2008.

[50] M. Grabisch and C. Labreuche. Fuzzy measures and integrals in MCDA. In José Figueira, Salvatore Greco, and Matthias Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, chapter 14, pages 563 – 609. Springer's International Series, 2005.

[51] M. Grabisch, J. Marichal, R. Mesiar, and E. Pap. *Aggregation Functions (Encyclopedia of Mathematics and Its Applications)*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[52] M. Grabisch, T. Murofushi, and M. Sugeno, editors. *Fuzzy Measures and Integrals: Theory and Applications*. Physica, 2000.

[53] M. Grabisch and J. M. Nicolas. Classification by fuzzy integral: performance and tests. *Fuzzy Sets and Systems*, 65(2-3):255–271, 1994.

[54] M. Grabisch and M. Roubens. Application of the Choquet integral in multicriteria decision making. In Michel Grabisch, Toshiaki Murofushi, and Michio Sugeno, editors, *Fuzzy Measures and Integrals: Theory and Applications*, pages 348–374. Physica, 2000.

[55] S. Greco, B. Matarazzo, and R. Slowinsk. Decision rule approach. In *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 78 of *International Series in Operations Research & Management Science*, pages 507–555. Springer New York, 2005.

[56] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[57] T. Hofmann, B. Scholköpf, and A. J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36:1171–1220, 2008.

[58] D. Hosmer and S. Lemeshow. *Applied Logistic Regression*. Wiley, 2nd edition, 2000.

[59] E. Hüllermeier and A. Fallah Tehrani. On the VC-dimension of the choquet integral. In Salvatore Greco, Bernadette Bouchon-Meunier, Giulianella Coletti, Mario Fedrizzi, Benedetto Matarazzo, and RonaldR. Yager, editors, *Advances on Computational Intelligence*, volume 297 of *Communications in Computer and Information Science*, pages 42–50. Springer Berlin Heidelberg, 2012.

[60] E. Hüllermeier and A. Fallah Tehrani. Efficient learning of classifiers based on the 2-additive choquet integral. In Christian Moewes and Andreas Nürnberger, editors, *Computational Intelligence in Intelligent Data Analysis*, volume 445 of *Studies in Computational Intelligence*, pages 17–29. Springer Berlin Heidelberg, 2013.

[61] H. Imai, D. Asano, and Y. Sato. An algorithm based on alternative projections for a fuzzy measures identification problem. *Springer*, pages 149–159, 2003.

[62] H. Imai, M. Miyamori, M. Miyakosi, and Y. Sato. An algorithm based on alternative projections for a fuzzy measures identification problem. *Proceedings of International Conference on Soft Computing, CD-ROM*, 200.

[63] Y. Ishii. Identification of multiple Choquet integral models by ga. *Master Thesis*, Tokyo Institute of Technology, 2000.

[64] Y. Ishii and T. Murofushi. Identification of fuzzy measures using real valued ga and considering outliers. *Proceedings of 6th Workshop on Evaluation of Heart and Mind*, 2001.

[65] J. Jaccard. *Interaction Effects in Logistic Regression*, volume 07-135 of *Saga University Papers Series on Quantitative Applications in the Social Sciences*. Saga Publications, 2001.

[66] W. S. Jewell. Isotonic optimization in tariff construction. *ASTIN Bulletin*, 8:175 – 203, 1975.

[67] W. Kotłowski, K. Dembczyński, S. Greco, and R. Słowiński. Stochastic dominance-based rough set model for ordinal classification. *Information Sciences*, 178(21):3989–4204, 2008.

[68] W. Kotłowski and R. Słowiński. Rule learning with monotonicity constraints. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML 2009, pages 537–544, New York, NY, USA, 2009. ACM.

[69] N. Landwehr, M. Hall, and E. Frank. Logistic model trees. In *Proceedings of the 14th European Conference on Machine Learning*, pages 241–252. Springer, 2003.

[70] S. Lee, H. Lee, P. Abbeel, and A.Y. Ng. Efficient L_1 regularized logistic regression. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 401–408. AAAI, 2006.

[71] T.W. Mason. Economic decision-making in the virtual firm. In *Engineering and Technology Management, 1996. IEMC 96. Proceedings., International Conference on*, pages 362–365, Aug 1996.

[72] G.A. Mendoza and H. Martins. Multi-criteria decision analysis in natural resource management: A critical review of methods and new modelling paradigms. *Forest Ecology and Management*, 230(1 - 3):1 – 22, 2006.

[73] P. Miranda, E. F. Combarro, and P. Gil. Extreme points of some families of non-additive measures. *European Journal of Operational Research*, 174(3):1865 – 1884, 2006.

[74] P. Miranda and M. Grabisch. On vertices of the k-additive monotone core. In *Proc. IFSA / EUSFLAT 2009*, pages 1194–1199, Lisbon, Portugal, 2009.

[75] P. Miranda, M. Grabisch, and P. Gil. Axiomatic structure of k-additive capacities. *Mathematical Social Sciences*, 49:153:178, 2005.

[76] F. Modave and M. Grabisch. Preference representation by a Choquet integral: commensurability hypothesis. In *Proceedings of the 7th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 164–171. Editions EDK, 1998.

[77] T. Mori and T. Murofushi. An analysis of evaluation model using fuzzy measure and the Choquet integral. In *Proceedings of the 5th Fuzzy System Symposium*, pages 207–212. Japan Society for Fuzzy Sets and Systems, 1989.

[78] T. Murofushi and S. Soneda. Techniques for reading fuzzy measures (III): interaction index. In *Proceedings of the 9th Fuzzy Systems Symposium, Sapporo, Japan*, pages 693–696, 1993.

[79] T. Murofushi, M. Sugeno, and M. Machida. Non-monotonic fuzzy measures and the Choquet integral. *Fuzzy Sets and Systems*, 64(1):73–86, 1994.

[80] M. Nasiri and S. Berlik. Modeling of polyester dyeing using an evolutionary fuzzy system. In Joao Paulo Carvalho, Didier Dubois, Uzay Kaymak, and Joao Miguel da Costa Sousa, editors, *Proceedings of the Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference*, pages 1246–1251. IFSA/EUSFLAT, 2009.

[81] A. S. Nemirovski and M. J. Todd. Interior-point methods for optimization. *Acta Numerica*, 17(-1):191–234, 2008.

[82] S. Opricovic and G-H. Tzeng. Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. *European Journal of Operational Research*, 156:445–455, 2004.

[83] P. M. Pardalos and G. Xue. Algorithms for a class of isotonic regression problems. *Algorithmica*, 23(3):211–222, 1999.

[84] M. Pirlot and H. Schmitz. An empirical comparison of the expressiveness of the additive value function and the Choquet integral models for representing rankings. In *URPDM - 2010, Mini- EURO Conference Uncertainty and Robustness in Planning and Decision Making, Coimbra, Portugal,*, 2010.

[85] R. Potharst and A. Feelders. Classification trees for problems with monotonicity constraints. *ACM SIGKDD Explorations Newsletter*, 4(1):1–10, 2002.

[86] B. Roy. The outranking approach and the foundations of ELECTRE methods. *Theory and Decision*, 31(1):49–73, 1991.

[87] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2001.

[88] B. Schölkopf, A. Smola, and K. R. Müller. Kernel principal component analysis. In *Advances in kernel methods: Support vector learning*, pages 327–352. MIT Press, 1999.

[89] R. Senge and E. Hüllermeier. Top-down induction of fuzzy pattern trees. *Fuzzy Systems, IEEE Transactions on*, 19(2):241–252, 2011.

[90] L. Shapley. A value for n-person games. In H.W. Kuhn and A.W. Tucker, editors, *Contributions to the Theory of Games*, volume 2, 1953.

[91] J. Sill. Monotonic networks. In *Advances in Neural Information Processing Systems*, pages 661–667. The MIT Press, Denver, USA, 1998.

[92] E. Sperner. Ein Satz über Untermengen einer endlichen Menge. *Mathematische Zeitschrift*, 27(1):544–548, 1928.

[93] Q. F. Stout. An approach to computing multidimensional isotonic regressions.

[94] Q. F. Stout. Strict L_∞ isotonic regression. *Journal of Optimization Theory and Applications*, 152(1):121–135, 2012.

[95] M. Sugeno. *Theory of Fuzzy Integrals and its Application*. PhD thesis, Tokyo Institute of Technology, 1974.

[96] K. Tanabe. Penalized logistic regression machines: New methods for statistical prediction 1. *ISM Cooperative Research Report 143*, pages 163 – 194, 2001.

[97] A. Fallah Tehrani, W. Cheng, K. Dembczyński, and E. Hüllermeier. Learning monotone nonlinear models using the Choquet integral. In *Machine Learning and Knowledge Discovery in Databases*, volume 6913 of *Lecture Notes in Computer Science*, pages 414–429. Springer Berlin Heidelberg, 2011.

[98] A. Fallah Tehrani, W. Cheng, K. Dembczyński, and E. Hüllermeier. Learning monotone nonlinear models using the Choquet integral. *Machine Learning*, 89(1-2):183–211, 2012.

[99] A. Fallah Tehrani, W. Cheng, and E. Hüllermeier. Choquistic regression: Generalizing logistic regression using the Choquet integral. In *Proceedings of the 7th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-2011) and LFA-2011*, volume 1 - 1 of *Advances in Intelligent Systems Research*, pages 868 – 875, July 2011.

[100] A. Fallah Tehrani, W. Cheng, and E. Hüllermeier. Preference learning using the Choquet integral: The case of multipartite ranking. *Fuzzy Systems, IEEE Transactions on*, 20(6):1102 –1113, dec. 2012.

[101] A. Fallah Tehrani and E. Hüllermeier. Ordinal choquistic regression. In *EUSFLAT Conf.* Atlantis Press, 2013.

[102] A. Fallah Tehrani, M. Strickert, and E. Hüllermeier. The Choquet kernel for monotone data. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, April 2014.

[103] R. J. Tibshirani, T. J. Hastie, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[104] V. Torra. Learning aggregation operators for preference modeling. In *Preference Learning*, pages 317–333. Springer, 2011.

[105] V. Torra and Y. Narukawa. *Modeling Decisions: Information Fusion and Aggregation Operators*. Springer, 2007.

[106] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[107] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[108] V. N. Vapnik. *Statistical Learning Theory*. A Wiley-Interscience Publication, October 1998.

[109] G. Vitali. Sulla definizione di integrale delle funzioni di una variabile. *Annali di Matematica Pura ed Applicata*, 2(1):111–121, 1925.

[110] A. Wald. Note on the consistency of the maximum likelihood estimate. *The Annals of Mathematical Statistics*, 20(4):595–601, 1949.

[111] N. Yan, Z. Wang, and Z. Chen. Classification with Choquet integral with respect to signed non-additive measure. *Seventh IEEE International Conference on Data Mining Workshops, ICDMW*, pages 283–288, 2007.

[112] Z. Yue. Extension of TOPSIS to determine weight of decision maker for group decision making problems with uncertain information. *Expert Systems with Applications*, 39:6343–6350, 2012.

[113] M. Zarghami, A. Abrishamchi, and R. Ardakanian. Multi-criteria decision making for integrated urban water management. *Water Resources Management*, 22(8):1017–1029, 2008.

[114] J. Zhu and T. Hastie. Support Vector Machines, Kernel Logistic Regression and Boosting. In *Proceedings of the Third International Workshop on Multiple Classifier Systems*, MCS '02, pages 16–26, London, UK, 2002. Springer-Verlag.

Erklärung

Ich versichere, dass ich meine Dissertation „Learning Nonlinear Monotone Classifiers Using The Choquet Integr“ selbständig, ohne unerlaubte Hilfe angefertigt und mich dabei keiner anderen als der von mir ausdrücklich bezeichneten Quellen und Hilfen bedient habe. Die Dissertation wurde in der jetzigen oder einer ähnlichen Form noch bei keiner anderen Hochschule eingereicht und hat noch keinen sonstigen Prüfungszwecken gedient.